

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Энергетический факультет


Кафедра Автоматизации производственных процессов и электротехники

Направление подготовки 15.03.04 – Автоматизация технологических процессов и производств

Направленность (профиль) образовательной программы Автоматизация технологических процессов и производств в энергетике

ДОПУСТИТЬ К ЗАЩИТЕ

И.о. зав. кафедрой


 О.В. Скрипко
« 26 » июня 2023 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка автоматизированной программы переключений для АСУТП подстанции Хани 220 кВ

Исполнитель

студент группы 941-об

20.06.23 
(подпись, дата)

Р.Е. Михолап

Руководитель

доцент, канд. техн. наук

 26.06.23
(подпись, дата)

А.Н. Рыбалев

Консультант:

по безопасности и
экологичности

доцент, канд. техн. наук

22.06.2023
(подпись, дата)

А.Б. Булгаков

Нормоконтроль

профессор, д-р. техн. наук

 26.06.2023
(подпись, дата)

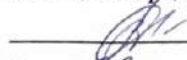
О.В. Скрипко

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический
Кафедра автоматизации производственных процессов и электротехники

УТВЕРЖДАЮ

И.о. зав.кафедрой

 О.В. Скрипко
« 26 » 06 2023 г

ЗАДАНИЕ

К выпускной квалификационной работе студента Михолапа Романа Евгеньевича

1. Тема выпускной квалификационной работы: Разработка автоматизированной программы переключений для АСУТП подстанции Хани 220 кВ

(утверждена приказом от _____ № _____)

2. Срок сдачи студентом законченной работы (проекта) 28.06.23 г.

3. Исходные данные к выпускной квалификационной работе: Рабочая документация ПС Хани 220 кВ, материалы собранные в ходе преддипломной практики

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): Описание объекта автоматизации, обзор подсистемы управления переключениями, разработка автоматизированной программы переключений

5. Перечень материалов приложения: (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.)

Лист 1: Однолинейная схема РУ 220 кВ ПС Хани;

Лист 2: Принципиальная схема ячейки подстанции и автоматизированная система управления переключениями;

Лист 3: Принципиальные электрические схемы управления приводами КА;

Лист 4: Схемы подключения к терминалу управления ЭКРА 242 шкафа УСО№1;

Лист 5: Пользовательские интерфейсы автоматизированной программы переключений;

Лист 6: Организационная структура автоматизированной программы переключений;


6. Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов) по безопасности и экологичности доцент, канд. техн. наук., Булгаков Андрей Борисович

7. Дата выдачи задания 10.04.2023 г.

Руководитель выпускной квалификационной работы: Рыбалёв Андрей Николаевич доцент канд. техн. наук

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению (дата, подпись): 26.05.2023 г.

(подпись студента) 

РЕФЕРАТ

Бакалаврская работа содержит 118 с., 27 рисунков, 2 таблицы, 7 приложений, 16 источников.

CODESYS, АСУТП, ТЕХНОЛОГИЧЕСКИЕ СРЕДСТВА АВТОМАТИЗАЦИИ, ПС ХАНИ 220 КВ, ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС, КОММУТИРУЮЩИЕ АППАРАТЫ, АПП, ФУНКЦИОНАЛЬНЫЙ БЛОК.

В работе исследованы коммутирующие аппараты подстанции, схемы их подключения к терминалам управления, изучена система управления аппаратами.

Цель работы - разработка автоматизированной программы переключений для АСУ ТП ПС Хани 220 кВ.

Методологию разработки составила итерационная модель и объектно-ориентированное программирование.

На основании данных методов была разработана автоматизированная программа переключений в программной среде CODESYS 3.5.

СОДЕРЖАНИЕ

Введение	7
1 Подстанция Хани 220 кВ	8
1.1 Оборудование	8
1.1.1 Коммутирующие аппараты и трансформаторы подстанции	8
1.1.2 Оборудование АСУ ТП подстанции	9
2 Система управления коммутирующей аппаратурой	12
2.1 Контроллер среднего уровня Advantech ECU-4784-D5SBE	12
2.2 Терминал управления присоединением ЭКРА 242	16
3 Подсистема управления переключениями	22
3.1 Общие требования к АПП	22
3.2 Управление КА на подстанции Хани 220 кВ	23
3.2.1 Разъединитель	25
3.2.2 Выключатель на ВЛ 220 кВ Юктали	32
3.2.3 Выключатель на линии трансформатора	37
4 Разработка автоматизированной программы переключений	44
4.1 Подсистема формирования и хранения режима переключений	45
4.1.1 Интерфейсы подсистемы	45
4.1.2 Программный код подсистемы	49
4.2 Подсистема выполнения плана переключений	54
4.2.1 Структуры программы	54
4.2.2 Функциональные блоки программы	57
4.2.3 Компоненты организации программы (POU)	68
5 Безопасность и экологичность	72
5.1 Безопасность	72
5.1.1 Меню формирования режима переключений	72
5.1.2 Меню управления	75
5.2 Экологичность	76
5.3 Чрезвычайные ситуации	78
Заключение	80
Библиографический список	81
Приложение А	83

Приложение Б	84
Приложение В	85
Приложение Г	86
Приложение Д	87
Приложение Е	88
Приложение Ж	89

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

- АВР – автоматический ввод резерва;
- АПВ – автоматическое повторное включение;
- АПП – автоматизированная программа переключений;
- АРМ – автоматизированное рабочее место;
- АСУ ТП – автоматизированная система управления технологическим процессом;
- АУВ – автоматика управления выключателем;
- ВБУ – внутренний блок управления;
- ВЛ – воздушная линия;
- ДУ – дистанционное управление;
- ЗН – заземляющий нож;
- ИО – измерительный орган;
- КА – коммутирующая аппаратура;
- ЛР – линейный разъединитель;
- ОРУ – открытое распределительное устройство;
- ПС – подстанция;
- РПО – реле положения «отключено»;
- РУ – распределительное устройство;
- СВ – секционный выключатель;
- ТАПВ – трехфазное автоматическое повторное включение;
- УСО – устройство связи с объектом;
- ШНЭ – шкаф наружной установки;
- Э1 – элемент электрической сети №1;
- Э2 – элемент электрической сети №2;
- ЭМВ – электромагнит включения;
- ЭМО – электромагнит отключения.

ВВЕДЕНИЕ

На многих промышленных предприятиях управление технологическим процессом осуществляется посредством многочисленных аппаратов, выполняющих те или иные коммутации потоков энергии. Типичным примером является электрическая подстанция, на распределительных устройствах которой производятся различные переключения линий, шин, силовых и измерительных трансформаторов и другого оборудования. Основными видами коммутационных аппаратов являются высоковольтные выключатели, разъединители и заземляющие ножи. Эти аппараты оснащены собственными приводами, которые допускают как местное, так и дистанционное управление. В последнем случае команды управления в виде дискретных сигналов поступают от специализированного контроллера, соединенного посредством тех или иных каналов связи с оборудованием диспетчерского пункта. В процессе управления контроллер получает информацию о состоянии аппарата и режиме его функционирования.

Однако остается актуальной проблема формирования и автоматического выполнения переключений. Для решения данной проблемы была разработана автоматизированная программа переключений (АПП), задающая последовательность срабатывания аппаратов определенной группы.

Целью АПП является решение какой-либо более крупной задачи, например, отключения одной из линий электроснабжения и включения другой (резервной). В данной работе предложен подход к решению этой проблемы, основанный на использовании технологии объектно-ориентированного программирования, которая все более активно применяется в промышленной автоматизации.

1 ПОДСТАНЦИЯ ХАНИ 220 КВ

ПС 220 кВ Хани является трансформаторной подстанцией. Данная подстанция позволяет необходимым образом распределять электроснабжения различных объектов, таких видов как сельский, поселковый, городской и промышленный.

ПС 220 кВ Хани расположена в Нерюнгринском районе Республики Саха (Якутии). Год постройки данной ПС – 1989. Ввод в эксплуатацию состоялся в 1989 году.

РУ ВН выполнено по схеме «Мостик с выключателями в цепях трансформаторов и ремонтной перемычкой со стороны трансформаторов». Тип РУ ВН – ОРУ.

В данной работе подробно рассматривается РУ ВН.

К РУ ВН подходят:

- 1) ВЛ 220 кВ Хани-Чара;
- 2) ВЛ 220 кВ Юктали-Хани с отпайкой на ПС Олёкма.

Данная ПС суммарная установленная мощность, которой составляет 50 МВА осуществляет прием, распределение и передачу электроэнергии на напряжениях 220, 35, 10 кВ.

Однолинейная схема РУ 220 кВ ПС 220 Хани с указанием точек ТИ, ТС, ТУ указана в Приложении А.

1.1 Оборудование

1.1.1 Коммутирующие аппараты и трансформаторы подстанции

На ПС 220 кВ Хани установлено следующее оборудование:

1. Силовые трансформаторы Т-1 и Т-2 марки – ТДТН-25000/220/35/10-70У1. Трансформатор Т-1 изготовлен в 1987 г., Т-2 изготовлен в 1988 году, оба трансформатора введены в эксплуатацию в 1989 году. На Т-1 установлено устройство РПН типа – У РС-4-74, на Т-2 тип РПН – ТУ РС-9-85. Установлено устройство ПБВ. Место установки 35 кВ, У.

2. ОПН тип – ОПНп-220/800/154-10-III УХЛ1, произведено и введено в эксплуатацию в 2007 году.

3. Шунтирующие реакторы: Р-1, Р-3, Р-4, Р-5, Р-6. Тип – РТМ-3300/10 У1. Изготовлены и введены в эксплуатацию в 1989 году.

4. Выключатели: секционный выключатель (СВ) и выключатели Т-1, Т-2. СВ типа GL-314, произведены AREVA в 2010 и 2009 гг. соответственно. Введены в эксплуатацию в 2011 и 2010 гг. Тип привода – FK3-1.

Выключатели Т-1 и Т-2 типа HPL-245 В1, произведен АBB в 2010 году, введен в эксплуатацию в этом же году. Тип привода – BLG 1002А.

Выключатель Юктали типа ВГТ-220-1К., введён в эксплуатацию в 2010 году. Тип привода - BLG 1002А.

5. Разъединители – 12х РГН-2-220/1000, введены в эксплуатацию в 2006 году. Привод – ПД-14-УХЛ1.

6. Трансформаторы тока – ТФЗМ-22 Б-III-У1 (яч. 4 ф. А, С), ТФЗМ-220 Б-III-ХЛ1 (яч. 4 ф. В; яч. 6 ф. А, В; яч. 5 ф. А, С), ТФНД-220-IV (яч. 6 ф. С, яч. 5 ф. В), ТОГФ-220-III-УХЛ1 введены в эксплуатацию в 2001-2003 гг.

7. Трансформаторы напряжения – НКФ-220-58 У1, изготовлены в 2007 году, введены в эксплуатацию в 2011 году. НАМИ-220, изготовленные в 1987 г., введённые в эксплуатацию в 1989 году.

1.1.2 Оборудование АСУ ТП подстанции

1. Управляемый коммутатор сети Ethernet – 8х коммутатор Ruggedcom RSG2300NC-R-RM-НН-НН-FX. 2х Коммутатор Ruggedcom RSG2100NC-R-RM-НН-НН-ТХ. Предназначены для надежной работы в окружении сильного электромагнитного излучения и климатически сложных условиях на сетевых подстанциях, в железнодорожных и промышленных зонах.

2. Сервер точного времени – 2х СВ-04. Устройство предназначено для формирования и хранения шкалы времени (ШВ), синхронизированной по сигналам глобальных навигационных спутниковых систем (ГНСС) ГЛОНАСС/GPS с национальной шкалой координированного времени Российской Федерации UTC(SU), а также выдачи информации о текущем значении даты и времени.

3. Цифровой измерительный преобразователь – 12х ЭНИП-2 устройство осуществляет измерение параметров режимов электрических сетей переменного трехфазного тока с номинальной частотой 50 Гц, индикацию синхронизированных векторных измерений, выполняют функций телеуправления, телесигнализации и технического учета электроэнергии с обеспечением обмена информацией по гальванически развязанным цифровым интерфейсам RS-485 и/или Ethernet. ЭНИП-2 предназначены для применения в составе систем сбора и передачи информации трансформаторных подстанций, распределительных пунктов, электростанций. Также они позволяют создавать распределенные системы телемеханики, системы технического учета электроэнергии, системы мониторинга качества электрической энергии, системы мониторинга переходных режимов.

4. Терминал ЭКРА 242 – 10х Терминал управления присоединением предназначен для организации управления коммутационным оборудованием электрических ПС в соответствии с группой стандартов МЭК 61850.

Терминалы управления присоединением осуществляют:

а) сбор и обработку аналоговой и дискретной информации по присоединению от блок-контактов первичного оборудования, контактов реле, датчиков и преобразователей;

б) формирование команд управления КА, РПН и другими аппаратами;

в) программную оперативную блокировку управления КА присоединения;

г) обмен информацией с другими терминалами управления, РЗА и ПА, с устройствами ССПИ/АСУ ТП среднего уровня с использованием протокола МЭК 61850-8-1;

д) резервное управление КА при неисправности средств верхнего или среднего уровней.

5. Коммутатор RedBox - Ruggedcom RSG909R 6GK6498-0RB00-1CN0-Z. полностью управляемый промышленный коммутатор Ethernet со встроенным HSR/PRP RedBox, предназначенный для критически важных, чувствительных ко времени устройств (например, подстанций МЭК 61850), которые не предусматривают простой.

6. Сервер асинхронных портов – 2х Сервер портов NPort 5650-8-DT-J компании Моха. Предназначен для подключения любых устройств с последовательным интерфейсом RS-232 или RS-422/RS-485 к Ethernet сетям, для построения систем безопасности, диспетчеризации, сбора данных и т.п.

7. Контроллер среднего уровня – 2х Advantech ECU-4784-D5SBE. Основан на IEC 61850-3, стандарте проектирования автоматизации электрических подстанций, который определяет связь между всеми интеллектуальными электронными устройствами (IED) и другим соответствующим оборудованием от уровня процесса (сбор данных, датчики и приводы) до уровня присоединения (защита), задачи мониторинга и управления и вплоть до уровня станции.

2 СИСТЕМА УПРАВЛЕНИЯ КОММУТИРУЮЩЕЙ АППАРАТУРОЙ

В данной системе управления достаточно будет рассмотреть работу и взаимосвязь двух ключевых контроллеров управления: Advantech ECU-4784-D5SBE и терминал управления присоединением ЭКРА 242.

Взаимосвязь контроллеров осуществляется по основной сети PPR A, Ethernet RJ 45/FO через управляемый коммутатор Ruggedcom RSG2300NC и т.к. разработанная программа написана для АРМ АСУ и АРМ ОП, то именно с них приходит сигнал на включение/отключение контроллеру Advantech ECU-4784-D5SBE и после выполнения всех операций, контроллер посылает сигнал на коммутатор Ruggedcom RSG2300NC откуда сигнал идёт на определённый терминал управления ЭКРА 242 (выбор терминала осуществляется контроллером). ЭКРА 242 посылает сигнал на включение/отключение подключённым к ней коммутирующим аппаратам.

Работа аппаратов рассмотрена далее в отдельных подразделах.

2.1 Контроллер среднего уровня Advantech ECU-4784-D5SBE

Контроллер определяет связь между всеми интеллектуальными электронными устройствами (IED) и другими связанными с ним устройствами, начиная с уровня процесса (сбор данных, датчики, исполнительные механизмы) и заканчивая задачами защиты, мониторинга и управления на уровне присоединения и на уровне станции (HMI/SCADA). Контроллер также действует как интеллектуальный коммуникационный сервер уровня подстанции; рабочей станции HMI/SCADA; станции удаленного управления; сервера сетевой безопасности (UTM) и станция мониторинга окружающей среды, позволяя системным интеграторам оптимизировать цифровые системы автоматизации подстанций и позволяя коммунальным предприятиям поставлять электроэнергию более безопасно, надежно, экономично и эффективно [8]. Внешний вид контроллера представлен на рисунках 1 и 2.



Рисунок 1 – Вид спереди ECU-4784-D5SBE



Рисунок 2 – Вид сзади ECU-4784-D5SBE

Серия ECU-4784 от Advantech сертифицирована по IEC 61850-3 и IEEE 1613 для соответствия жестким и критическим требованиям автоматизации интеллектуальных сетей.

Он обеспечивает высокую производительность и мощность для наиболее важных вычислительных приложений для решений виртуализации, снижая затраты на развертывание и обслуживание проекта.

Осуществляет поддержку удаленного мониторинга и диагностики подключений удаленных устройств для повышения эффективности управления и обслуживания.

Технические характеристики оборудования [8]:

1) Сертификация: IEC 61850-3, IEEE 1613, CE, FCC класс A, UL, CCC, CB, LVD;

- 2) Размеры (Ш x Д x В): 2U (440 x 280 x 88 мм / 17,3» x 8,6» x 3,4») стандартная 19-дюймовая стойка;
- 3) Корпус: SECC и алюминий;
- 4) Монтаж: монтаж в стойку 2U;
- 5) Потребляемая мощность: 19 Вт/220 В переменного или постоянного тока;
- 6) Требования к питанию: 100 – 240 В переменного тока или 100 – 240 В постоянного тока;
- 7) Тип питания: ATX, AT;
- 8) Вес: 6,0 кг;
- 9) Конструкция системы: конструкция без вентилятора;
- 10) Поддержка ОС: Windows7, Windows8, Window server 2008 R2; Windows server 2012R2, Windows Embedded8 (64bits), Linux;
- 11) iCDManager: Интеллектуальное Программное Обеспечение для диагностики и управления подключением;
- 12) Центральный процессор: Intel Haswell Cor i7;
- 13) Память: DDR3L 1,35 В без ECC 8 ГБ (до 16 ГБ на 2 модуля 8 ГБ);
- 14) Индикаторы: светодиоды питания, жесткого диска, программируемый светодиод, LAN (активный, положение дел) и серийный (Tx, Rx);
- 15) Хранилище: 2 жестких диска SATA 2,5» (RAID 0,1); 1 разъем CFast;
- 16) Дисплей: разъем VGA DB15, 1 x DVI-D, 1920 x 1200, 60 Гц;
- 17) Сторожевой таймер: программируемый 256-уровневый, временной интервал от 1 до 255 секунд для каждого уровня (чип Super I/O: W83627HG);
- 18) Последовательные порты: а) 2 x DB-9 RS-232 (5-контактный); б) 8 x винтовых клемм с 3-проводным интерфейсом RS-422/485 (клеммная колодка) (Автоматическое управление потоком данных RS-485) Изоляция 2000 В постоянного тока;
- 19) Скорость последовательного порта:
 - (COM1, COM2) RS-232: 50 ~ 115200 бит/с,
 - (COM3 ~ COM8) RS-422: 50 ~ 115200 бит/с, RS-485: 50 ~ 921600 бит/с;

20) LAN: а) 1 х порт 10/100/1000 Base T RJ45 (контроллер Intel i218LM Gigabit Ethernet, АМТ, функция объединения, PXE); б) 7 х портов 10/100/1000 Base T RJ45 (контроллер Intel i210 Gigabit Ethernet, функция объединения, PXE);

21) Порты USB: 6 х USB, UHCI, совместимость с Rev. 2.0, 2 спереди, 3 сзади и 1 внутренний порт;

22) Расширение: 2 расширения ввода/вывода домена (каждый слот поддерживает до 1 PCIe или 2 PCI-устройства);

23) Релейный выход: форма С, контакт 5 А при 125 В переменного тока/5 А при 30 В постоянного тока (для IEC-61010-2-201);

24) Программируемый светодиод: 8-канальный программируемый светодиодный индикатор;

Для более чёткого понимания устройства контроллера Advantech ECU-4784-D5SBE на рисунке 3 приведена схема его функционального блока.

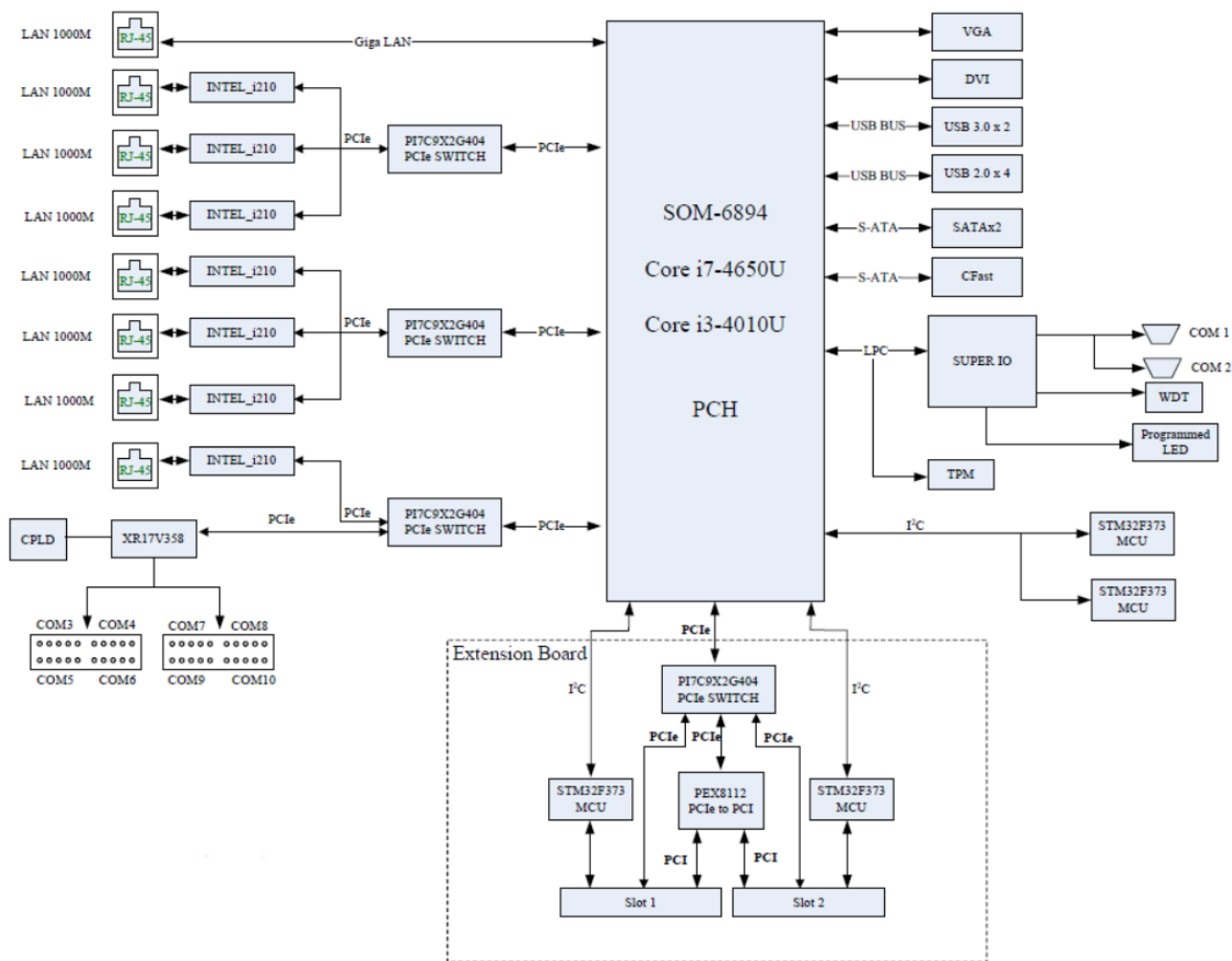


Рисунок 3 – Функциональный блок ECU-4784-D5SBE

2.2 Терминал управления присоединением ЭКРА 242

Терминал управления присоединением типа ЭКРА 242 – микропроцессорное устройство, предназначенное для управления выключателем и коммутационными аппаратами присоединения, организации оперативных блокировок, сбора и обработки аналоговой и дискретной информации, измерения электрических параметров сети переменного тока и/или силы постоянного тока [7].

Данные терминалы управления устанавливаются в шкафах УСО (шкаф устройств сопряжения с объектом). Шкаф УСО с терминалом ЭКРА 242 образует устройство ШНЭ 2090. ШНЭ 2090 - типовой шкаф наружной установки для подстанций для сопряжения с объектом по протоколам стандарта МЭК 61850-8-1. Данное устройство предназначено для сбора и преобразования дискретной информации о текущих технологических режимах и состоянии первичного оборудования и реализации управляющих воздействий на него. Устройства устанавливаются в непосредственной близости от первичного оборудования. На подстанции Хани 220 кВ имеется 7 штук ШНЭ 2090, каждый из них управляет своей ячейкой. Внешний вид терминала представлен на рисунке 4.

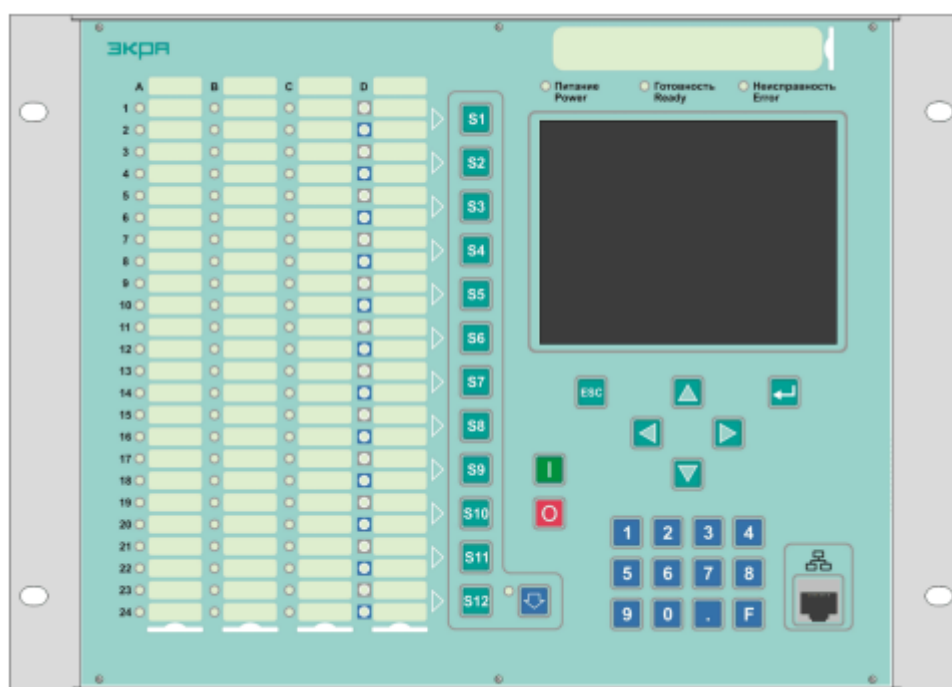


Рисунок 4 – Внешний вид терминала управления присоединением ЭКРА 242

Терминал управления присоединением типа ЭКРА 242 может выполнять следующие функции:

- а) Автоматика управления выключателем (АУВ):
 - 1) для выключателей (6 - 35) кВ с трехфазным приводом;
 - 2) для выключателей 110 кВ и выше с трехфазным приводом;
 - 3) для выключателей 110 кВ и выше с пофазным приводом.
- б) Автоматический ввод резерва (АВР):
 - 1) на пуск АВР;
 - 2) на включение выключателя от АВР.
- в) Автоматика восстановления нормального режима;
- г) Автоматическое повторное включение (АПВ):
 - 1) ускоренное трехфазное автоматическое повторное включение;
 - 2) трехфазное автоматическое повторное включение с контролем отсутствия напряжения;
 - 3) трехфазное автоматическое повторное включение без контроля напряжений;
 - 4) несинхронное трехфазное автоматическое повторное включение;
 - 5) трехфазное автоматическое повторное включение с контролем синхронизма;
 - 6) однофазное автоматическое повторное включение.
- д) Устройство резервирования при отказе выключателя:
 - 1) с компенсацией емкостного тока линии;
 - 2) с пуском от защит стороны низшего напряжения трансформатора при недостаточной чувствительности измерительного органа (ИО) тока УРОВ при коротких замыканиях на низшей стороне трансформатора;
 - 3) с ускорением действия при низком давлении газа в баке выключателя.
- е) Синхронизация генератора с энергосистемой:
 - 1) автоматическая синхронизация;
 - 2) полуавтоматическая синхронизация;

- 3) ручная синхронизация.
- ж) Автоматика регулирования коэффициента трансформации;
- и) Измерение электрических параметров сети переменного тока и/или силы постоянного тока;
- к) Управление коммутационными аппаратами присоединения:
 - 1) через цифровые каналы связи;
 - 2) через мнемосхему терминала с отображением состояний коммутационных аппаратов присоединения.
- л) Выполнение пользовательских алгоритмов (ОБ и др.);
- м) Контроль ресурса коммутационных аппаратов.

Т.к. на данной подстанции управление выключателями осуществляется при помощи АУВ, то рассмотрим эту функцию более подробно.

В дальнейшем сокращения «Э1» и «Э2» означают элемент сети 1 или 2 (шина, линия, трансформатор, автотрансформатор).

Автоматика управления выключателем содержит следующие узлы и защиты [7]:

- 1) Узел включения выключателя;
- 2) Узел отключения выключателя;
- 3) Защита электромагнитов управления от длительного протекания тока;
- 4) Узел контроля исправности цепей электромагнитов управления;
- 5) Узел фиксации положения выключателя;
- 6) Узел фиксации команд;
- 7) Узел фиксации автоматического отключения;
- 8) Узел РПО для фиксации отключения выключателя;
- 9) Защита от непереключения фаз;
- 10) Защита от неполнофазного режима;
- 11) Узел дистанционного управления выключателем и управления с мнемосхемы терминала;
- 12) Узел полуавтоматического оперативного включения с контролем отсутствия напряжения или с контролем синхронизма;

- 13) Узлы обработки входов внешнего отключения;
- 14) Узел запрета ТАПВ от внешнего сигнала;
- 15) Узел запрета ТАПВ Э1 или Э2 с противоположной стороны;
- 16) Узел ускорения защит.

Теперь рассмотрим работу узла включения и отключения АУВ.

Узел включения формирует сигналы для действия на электромагниты включения (ЭМВ).

Сигналы на выходе узла формируются при отсутствии блокирующих сигналов от дискретных входов «Пружины не заведены» или «Блокировка управления» и формировании на входе любого из следующих сигналов:

- 1) Сигнал на включение выключателя от оперативного ключа управления, через дискретный вход терминала;
- 2) Сигнал на полуавтоматическое включение от оперативного ключа, с узла оперативного включения с контролем условий
- 3) Сигнал на включение через цифровые каналы связи или с мнемосхемы терминала;
- 4) Сигнал на включение от АПВ.

Сигнал от дискретного входа «Блокировка управления» служит для блокировки включения выключателя от внешнего сигнала, например, при аварийном давлении элегаза.

Если на входе узла сигнал на включение возникает раньше сигнала «Блокировка управления», то узел продолжает действовать на включение, а блокировка управления обеспечивается после успешного завершения операции.

Для предотвращения неполного завершения или срыва операции включения предусмотрен подхват действия в цепи ЭМВ в течение всего времени, пока через электромагниты протекает ток. Подхват сигналов включения осуществляется с помощью датчиков тока ЭМВ, установленных в цепи электромагнитов выключателя.

Для реализации подхвата сигналов включения предусмотрены измерительные органы (далее - ИО) постоянного тока, для контроля постоянного тока цепей электромагнитов.

ИО постоянного тока определяет абсолютное значение тока и срабатывает, если это значение превышает уставку срабатывания. Средняя основная погрешность срабатывания ИО составляет не более 20 процентов от установленного значения срабатывания.

ИО имеет два выхода: один без задержки срабатывания и один с выдержкой времени на срабатывание.

Выход с задержкой активируется при превышении рабочей уставки в течение некорректируемой временной задержки «Тсраб.». Эта задержка предотвращает срабатывание ИО в случае скачка тока при неисправности цепи постоянного тока.

Она также предотвращает повторное замыкание автоматического выключателя в случае короткого замыкания (блокировка против отключения). Если вовремя сигнала замыкания в электромагните отключения обнаруживается протекание тока, то автоматический выключатель срабатывает один раз, а замкнутая цепь прерывается вовремя сигнала замыкания.

Узел отключения формирует сигналы для действия на электромагниты отключения (ЭМО).

Сигналы на выходе узла формируются при отсутствии сигнала «Блокировка управления» и при формировании любого из сигналов:

- 1) Сигнал на отключение выключателя от оперативного ключа управления, через дискретный вход терминала;
- 2) Сигнал на отключение через цифровые каналы связи или с мнемосхемы терминала;
- 3) Сигнал на отключение от внешних защит, с узлов обработки внешнего отключения;
- 5) Пуск УРОВ от защит низкой стороны трансформатора защищаемого присоединения;

- 6) Пуск УРОВ от защит смежных Э1 и Э2;
- 7) Срабатывание УРОВ «своего» выключателя;
- 8) Срабатывание защиты от непереключения фаз.

Сигнал от дискретного входа «Блокировка управления» служит для блокировки отключения выключателя, например, при аварийном давлении элегаза.

Для выключателя с пофазным приводом предусмотрено действие на пофазное отключение по сигналам срабатывания защит Э1 или Э2.

Для предотвращения неполного завершения или срыва операции отключения предусмотрен подхват цепей действия на электромагниты отключения на все время, пока по электромагнитам отключения протекает ток. Подхват сигнала отключения обеспечивается датчиком тока ЭМО, установленных в цепи электромагнитов выключателя. Для обеспечения подхвата сигналов отключения предусмотрены ИО постоянного тока.

В случае использования выключателей с двумя группами электромагнитов отключения (ЭМО1 и ЭМО2) подхват осуществляется при протекании тока хотя бы по одной из групп электромагнитов.

3 ПОДСИСТЕМА УПРАВЛЕНИЯ ПЕРЕКЛЮЧЕНИЯМИ

На каждой подстанции нового типа имеется автоматизированная программа переключений(далее-АПП). Именно с помощью неё происходит дистанционное управление подстанцией.

Автоматизированная программа переключений (АПП) – программа переключений, сформированная в электронном виде в автоматизированной системе диспетчерского управления диспетчерского центра или автоматизированной системе технологического управления центра управления сетями, центра управления ветровыми (солнечными) электростанциями на основании утвержденной программы (типовой программы) переключений, которая посредством программируемой логики выполняет переключения по изменению эксплуатационного состояния линий электропередачи и электросетевого оборудования, изменению технологического режима работы электросетевого оборудования и устройств релейной защиты и автоматики путем автоматической реализации блока операций с коммутационными аппаратами, заземляющими разъединителями (заземляющими ножами разъединителя) и устройствами релейной защиты и автоматики по заранее определенному алгоритму с проверкой возможности выполнения каждой операции (соответствие схемы, отсутствие блокирующих дистанционное управление сигналов аварийно-предупредительной телесигнализации, подтверждение выполнения предыдущей операции) [1].

3.1 Общие требования к АПП

1) АПП должны создаваться для конкретной исходной схемы электроустановки и при запуске посредством программируемой логики проверяться на наличие возможности выполнения переключений (соответствие схемы, отсутствие АПТС, подтверждение выполнения предыдущей операции и т.д.) [1];

2) Запуск АПП должен осуществляться путем выбора соответствующей АПП из перечня АПП;

3) Переключения с использованием АПП должно выполняться в соответствии с распределением функций технологического управления;

4) АПП должны создаваться в виде списка отдельных операций, создаваемого пользователем в ручном режиме путем выбора объектов управления на графической однолинейной схеме первичного оборудования и/или на структурной схеме ассоциированного вторичного оборудования и устройств, а также путем ручного ввода в данный список стандартных операций;

5) При выполнении АПП пользователь, инициирующий выполнение АПП, должен иметь возможность остановить выполнение АПП с последующим возобновлением выполнения АПП или отменой всех невыполненных операций АПП;

6) Должна быть реализована возможность автоматической остановки выполнения АПП при выполнении специальной, включенной в АПП операции, с последующим возобновлением выполнения АПП при подтверждении пользователем;

7) Перед запуском выбранной АПП должна выполняться проверка на наличие возможности выполнения АПП с формированием соответствующего сообщения пользователю;

8) Для каждой операции из АПП должно проверяться отсутствие блокировок доступа и состояния;

9) Для АПП в целом в режиме моделирования должна быть выполнена проверка отсутствия режимных блокировок;

10) Перед выполнением операции, входящей в состав АПП, должна выполняться проверка отсутствия блокировок управления;

11) Выполнение следующей операции АПП должно блокироваться до получения подтверждения о выполнении предыдущей операции;

12) Должны контролироваться результаты выполнения (состояния объектов управления) предыдущих операций АПП и в случае несоответствия заданному значению (состоянию объекта управления), выполнение АПП должно блокироваться.

3.2 Управление КА на подстанции Хани 220 кВ

В разделе 2 уже говорилось, что вся АСУ ТП подстанции управляется при помощи контроллера Advantech ECU-4784-D5SBE и терминалов управления

ЭКРА 242. Именно они формируют команды на управление коммутационной аппаратурой.

В данном разделе будет рассматриваться как осуществляется управление КА и их схемы подключения к терминалу управления ЭКРА 242.

Так как периферия подключения к терминалам ЭКРА 242, и тип всех разъединителей одинаковы на всех ячейках, то в качестве примера рассмотрим 7 ячейку подстанции – «ВЛ 220 кВ Юктали – Хани с отпайкой на ПС Олёкма, ТН-2-220».

На рисунке 5 представлен элемент принципиальной схемы ОРУ 220 кВ ПС Хани. На котором изображена 7 Ячейка подстанции Хани 220 кВ.

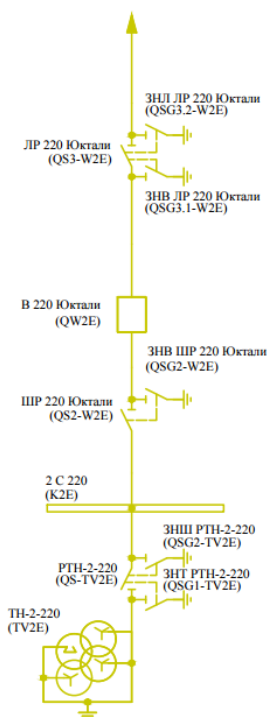


Рисунок 5 – Принципиальная схема. ВЛ 220 кВ Юктали – Хани с отпайкой на ПС Олёкма, ТН-2-220

Данным участком подстанции управляет Шкаф УСО №1. Его шкаф управления подключён к таким КА как: ЗНЛ ЛР 220 Юктали, ЛР 220 Юктали, ЗНВ ЛР 220 Юктали, В 220 Юктали, ЗНВ ШР 220 Юктали, ШР 220 Юктали, ЗНШ РТН-2-220, РТН-2-220, ЗНТ РТН-2-220.

Так как тип разъединителей и выключателей на всей подстанции схож, то далее схемы подключения к терминалу управления и принцип работы КА будут рассматриваться в отдельных пунктах.

3.2.1 Разъединитель

На рисунке 6 представлена схема подключения линейного разъединителя и одного из его заземляющих ножей к входам терминала управления ЭКРА 242.

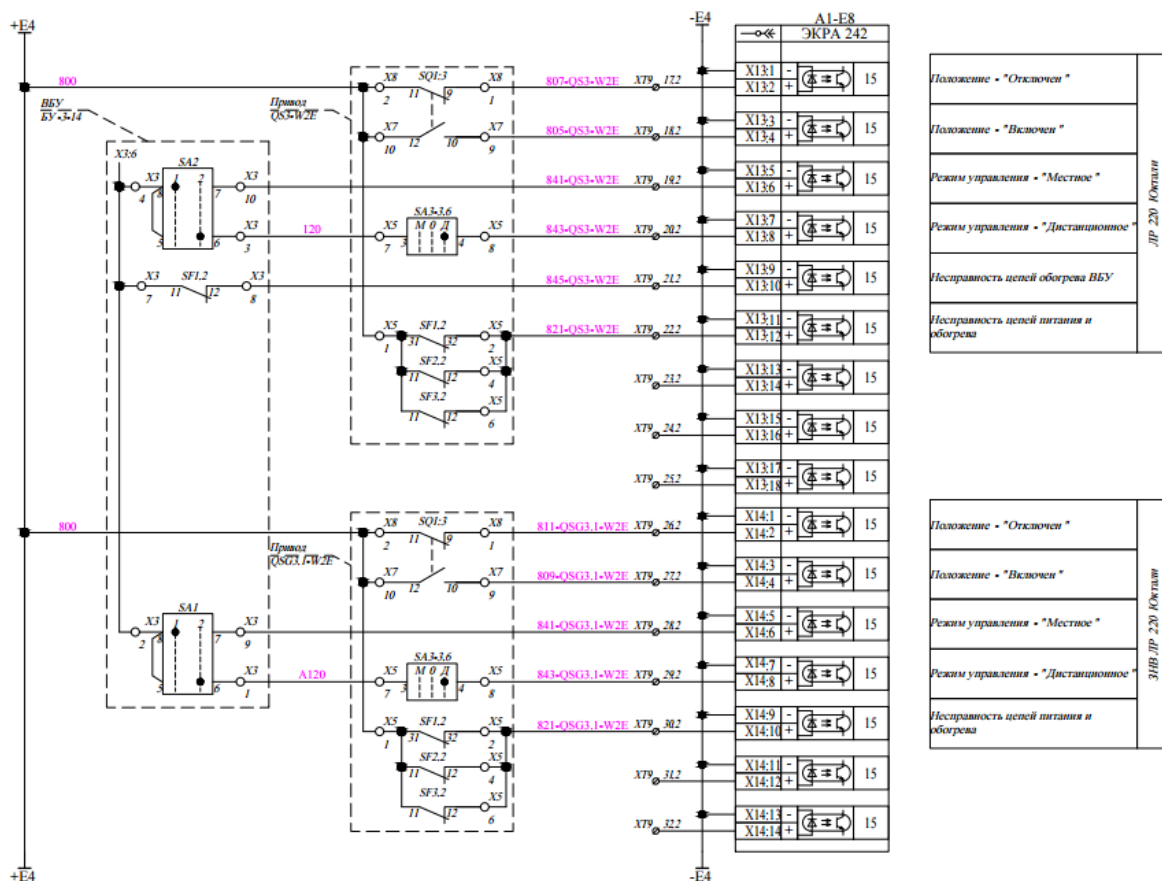


Рисунок 6 – Схема подключения ЛР и ЗНВ ЛР к терминалу управления ЭКРА 242

Так как заземляющий нож входит в устройство разъединителя, то рассмотрим только ЛР 220 Юктали. На схеме видно, что от данного аппарата приходят 6 сигналов:

- 1) Сигнал о положении привода линейного разъединителя – «отключен»;
- 2) Сигнал о положении привода линейного разъединителя – «включен»;
- 3) Сигнал о режиме управления ЛР – «Местное»;
- 4) Сигнал о режиме управления ЛР – «Дистанционное»;

5) Сигнал о неисправности цепей обогрева ВБУ (встроенный блок управления);

6) Неисправность цепей питания и обогрева самого линейного разъединителя;

Данный линейный разъединитель предназначен для включения и отключения обесточенных участков цепи высокого напряжения, токов холостого хода трансформаторов, зарядных токов воздушных линий, а также заземления отключенных участков при помощи стационарных заземлителей.

Линейный разъединитель 220 Юктали типа РГН.2-220.И/100-40УХЛ1.

Данный тип имеет не только линейный разъединитель, но также заземляющие ножи, шинные и трансформаторные разъединители, поэтому в данном пункте будут приведены характеристики и принцип работы всех разъединителей на подстанции. Его технические характеристики указаны в таблице 1 [3].

Таблица 1 - Технические характеристики РГН.2-220.И/100-40УХЛ1

Наименование параметра	Технические характеристики
1	2
Номинальное напряжение, кВ	220
Наибольшее рабочее напряжение, кВ	252
Номинальный ток, А	1000
Номинальный кратковременный выдерживаемый ток (ток термической стойкости), кА	40
Наибольший пик номинального кратковременного выдерживаемого тока (ток электродинамической стойкости), кА	100
Частота переменного тока, Гц	50
Время протекания номинального кратковременного выдерживаемого тока, с: - для главных ножей - для заземлителей	3 1
Испытательное одноминутное переменное напряжение в сухом состоянии и под дождем, кВ: - относительно земли и между полюсами;	440
- между разомкнутыми контактами одного и того же полюса	460
Испытательное напряжение грозового импульса 1,2/50 мкс, кВ:	

1	2
- относительно земли и между полюсами; - между разомкнутыми контактами разъединителей	900 1100
Допустимая механическая нагрузка на выводы, Н	1000
Сопротивление постоянному току главного токоведущего контура, Ом: не более	165×10^{-6}
Минимальная длина пути утечки внешней изоляции разъединителей для степени загрязнения изоляции, мм:	5070

Разъединитель выполнен в виде отдельных полюсов, представляющих собой двух колонковые аппараты с разворотом контактных ножей в горизонтальной плоскости. Внешний вид представлен на рисунках 7 и 8.

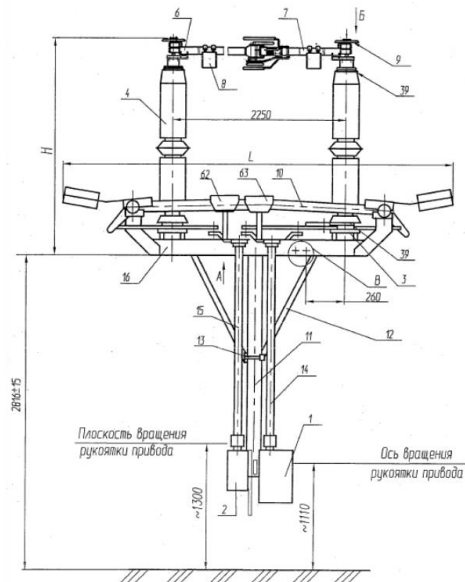


Рисунок 7 – Общий вид РГН.2-220.П/100-40УХЛ1:

1 - привод разъединителя типа ПД-14-00 УХЛ1; 2 - привод заземлителя типа ПРГ-01-6 УХЛ1; 3 - основание; 4 - изолятор; 6, 7 - ножи контактные; 8 - контакт заземляющего контура; 9 - вывод контактный; 10 - заземлитель; 11 - кранштейн; 12 - подкос; 13 - шпилька; 14 - вал разъединителя; 15 - вал заземлителя; 16 - цоколь; 39 - прокладка; 62, 63 – кожух.

Полюс разъединителя, к которому присоединяются приводы, называется ведущим.

Полюс разъединителя, присоединяемый к ведущему, называется ведомым.

Соединение ведущего полюса разъединителей с приводами (рисунок 7) и с ведомыми полюсами (рисунок 8) выполняется с помощью соединительных элементов, входящих в комплект поставки. Монтаж разъединителя осуществляется без применения сварки с помощью болтовых соединений.

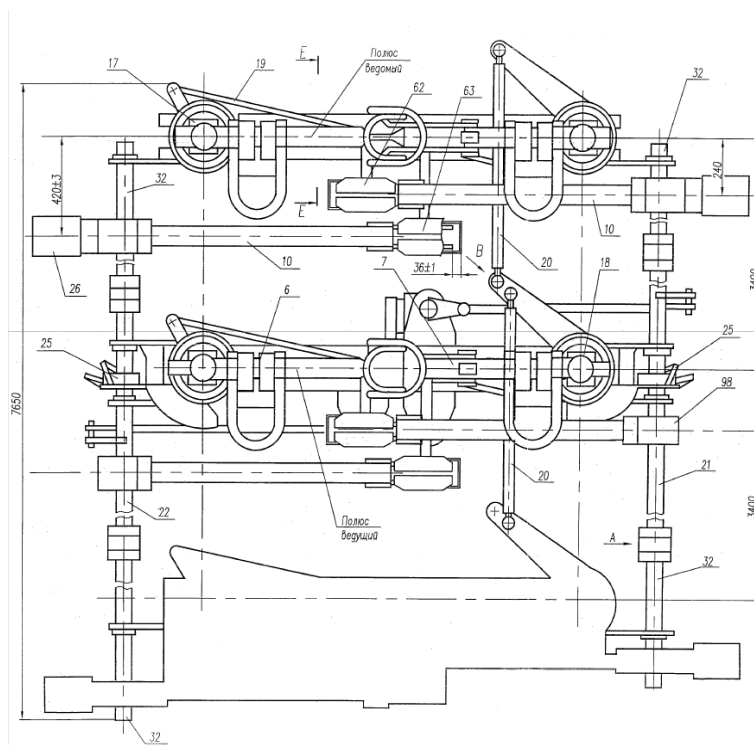


Рисунок 8 – Общий вид РГН.2-220.П/100-40УХЛ1:

6,7 - нож контактный; 10 - заземлитель; 17 - полюс ведомый; 18 - полюс ведущий; 19 - тяга внутрислолюсная; 20 - тяга межполюсная; 21, 22, 32 - вал; 25 - фиксатор; 26 - противовес; 62, 63 - кожух.

Каждый полюс разъединителя включает в себя систему возбуждения, состоящую из цоколя 16, изолятора 4 и контактных ножей 6 и 7 (рисунок 7).

Цоколь 16 (рисунок 8) каждого разъединителя состоит из двух швеллеров, в которых болтами закреплены два основания 3. Эти основания оснащены закрытыми шарикоподшипниками, которые смазываются в течение всего срока службы разъединителя.

В подшипниках вращаются валы, на которые устанавливаются изоляторы 4 и закрепляются рычаги передач.

Рычаги ведущего и ведомого изоляторов полюса разъединителей соединяются между собой регулируемой тягой 19 (рисунок 8).

К швеллерам цоколя крепятся кронштейны, в которые устанавливаются валы заземлителей 23, 24, 25 (рисунок 8).

Для крепления полюсов разъединителей к опорной металлоконструкции используются отверстия, разметка которых приведена на рисунке 6.

На ведущем цоколе расположены кронштейны для приводных валов главных ножей 14 и заземлителей 15 (рисунок 7).

Токоведущая система разъединителя выполнена в виде двух контактных ножей 6 и 7 (рисунок 7), которые устанавливаются на верхних фланцах изоляторов.

Каждый контактный нож состоит из основания, на котором крепятся алюминиевый нож и контактный вывод. Передача тока от медных шин к контактному выводу осуществляется через гибкое соединение.

На контактном ноже 7 (рисунок 7) имеется ламельный контакт, выполненный с использованием медных ламелей, не требующий регулировок контактного нажатия в эксплуатации в течение всего периода эксплуатации. Концы ламелей имеют отгибы (ловители) для обеспечения вхождения контакта противоположного ножа.

Ламельный контакт на ток 1000 А выполнен из двух пар контактных ламелей, на ток 2000 А из трех пар.

На конце ножа 6 (рисунок 7) имеется контакт типа «кулачок» с поверхностями контактирования, образованными отгибами параллельных шин. Контакт типа «кулачок» защищен от обледенения кожухом.

К контактными поверхностям разъёмного контакта припаяны серебряные накладки, все остальные контактные поверхности оловом.

На контактных ножах 6 и 7 (рисунок 7) размещены контакты заземляющих контуров. Контакты заземляющих контуров защищены от обледенения кожухами.

Трёхполюсная система разъединителей состоит из ведущего полюса 18 и двух ведомых полюсов 17, на которых монтируются заземлители 10 (рисунок 8).

На кронштейне, закрепленном на ведущем полюсе, устанавливаются приводы 1 и 2 (рисунок 7).

Заземлитель состоит из алюминиевой трубки, колодок, трех пар пластин. Пластины изготовлены из бериллиевого сплава и не требуют регулировки контактного нажатия в течение всего периода эксплуатации.

Шток заземлителя через гибкие связи соединяется с кронштейном 48 полюса ведущего полюса разъединителя. Разметка отверстий для крепления шины заземления приведена на рисунке 7.

Когда система привода главных контактных ножей 1 (рисунок 7) находится в работе, ведущий полюс поворачивает рычаг ведущего изолятора на угол 90° . В пределах рычага 19 полюса, который соединяет главный изолятор с рычагом ведомого изолятора, рычаг ведомого изолятора также поворачивается на угол 90° . Когда рычаг ведущего изолятора ведущего полюса поворачивается на 90° , межполюсный рычаг 20 (рисунок 8) поворачивает рычаги ведущих изоляторов ведомых полюсов на тот же угол.

Под действием этого рычага изоляторы 4 ведущей и ведомой штанг и закрепленные на них контактные элементы 6 и 7 (рисунок 7) поворачиваются на 90° , замыкая цепь.

При включении привода заземления рычаг вызывает поворот штанги и заземлителей, установленных на валах 23, 24 и 25 (рисунок 8), на угол 76° , в этот момент контакт ножей заземлителя пересекает контакт главных контактных ножей 6 и 7 (рисунок 7).

После разбора устройства и работы самого разъединителя необходимо рассмотреть работу его привода. Привод разъединителя типа ПД-14-00 УХЛ1.

Схема подключения к приводу представлена на рисунке 9.

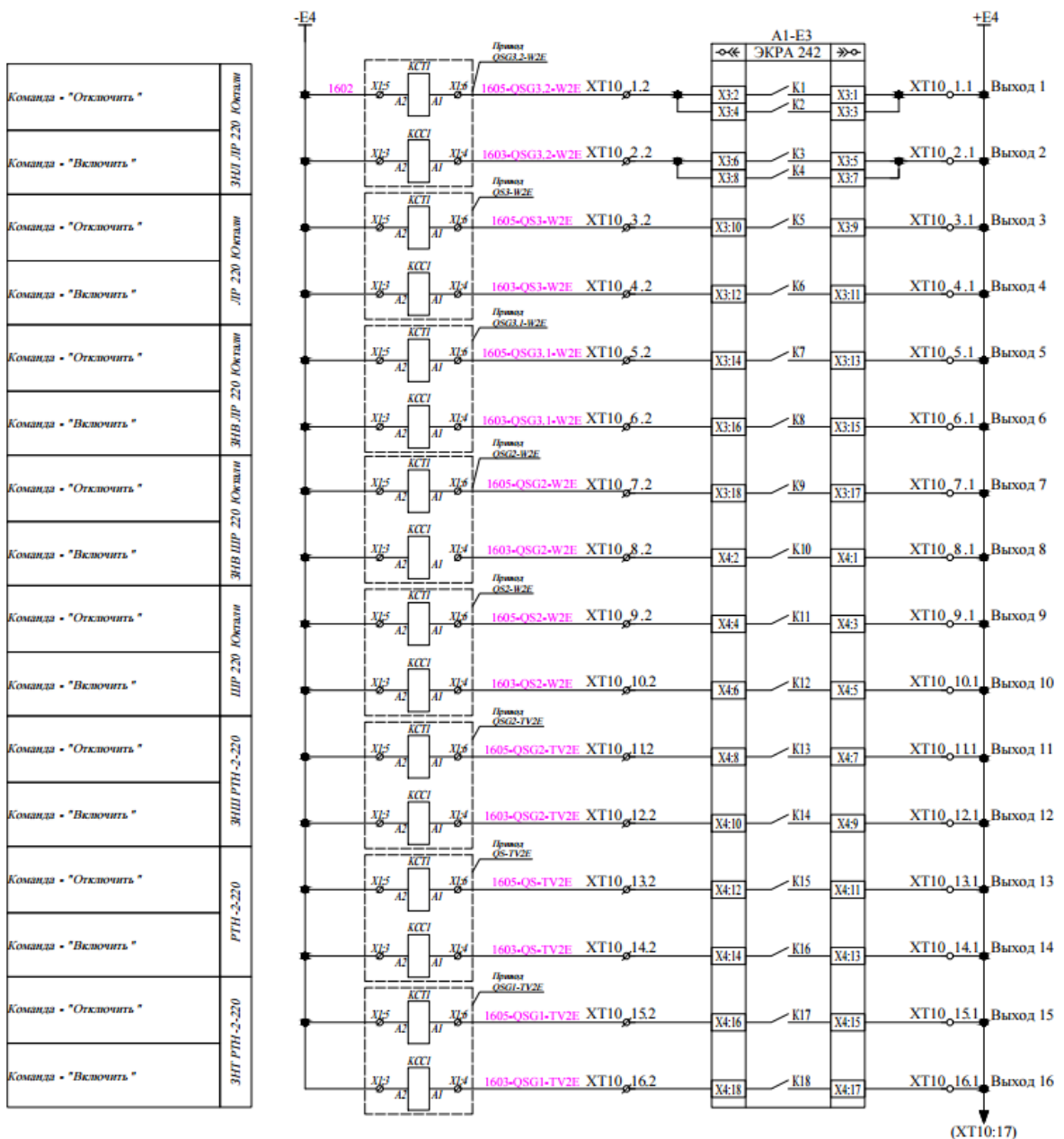


Рисунок 9 – Схема подключения терминала управления к катушкам привода

Принципиальная электрическая схема привода со всеми обозначениями приведена в приложении В.

Управление электродвигателем привода осуществляется нажатием кнопок SB1 или SB2 (местное управление) и с центрального пульта путём подачи электрических сигналов управления через кабель управления на релейные катушки KCC или KCT (дистанционное управление). При обоих типах управления двигателем (местном или дистанционном) управление приводом возможно только при подаче питания на катушку электромагнитной блокировки YA [3].

В электрическую схему включен переключатель режима управления местное-дистанционное (переключатель SA). В положении «дистанционное» блокируются кнопки управления SB1 и SB2, а в положении «местное» блокируются контакты реле КСС и КСТ. Привод имеет возможность электрической блокировки, которая означает, что выключатель заземления не может быть задействован, если главный выключатель разомкнут. Электрическая блокировка достигается путем удаления перемычки с X15 - X16 в ряду клемм и вставки в этот разрыв соответствующих вспомогательных контактов приводного блока.

Сигнальные лампы VDR и VDG, указывающие на положение разъединителя или заземлителя, являются встроенными в клапан светодиодами и включаются контактами концевых микровыключателей SQC и SQT. В конце работы механизм *malti* поворачивает вал выключателя FC на 90 градусов, и контакты переключателя переключаются. Лампа VDR (красная) указывает на активное положение, а лампа VDG (зеленая) - на неактивное положение.

3.2.2 Выключатель на ВЛ 220 кВ Юктали

На рисунке 10 представлено подключение выключателя и одного из заземляющих ножей линейного разъединителя.

Заземляющий нож линейного разъединителя был рассмотрен в подразделе 3.2.1. поэтому в данном подразделе будет рассматриваться выключатель – В 220 Юктали.

На схеме видно, что от данного аппарата приходит 2 сигнала:

- 1) Сигнал о положение выключателя – «отключен»;
- 2) Сигнал о положение выключателя – «включен»;

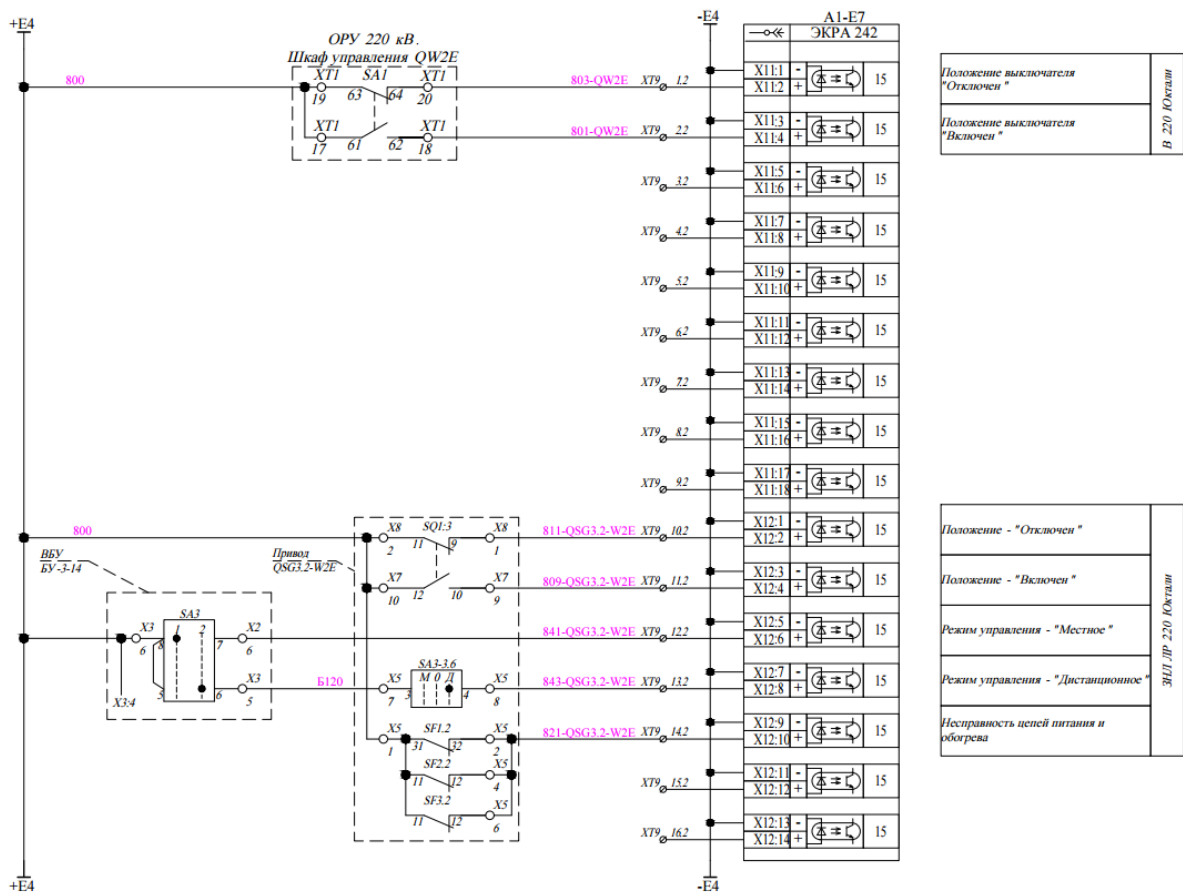


Рисунок 10 – Схема подключения выключателя и ЗНЛ ЛР к терминалу управления ЭКРА 242

Данный элегазовый выключатель типа ВГТ-220-1К УХЛ1, 4000 А, 40кА. Его технические характеристики приведены в таблице 2 [5].

Таблица 2 - Технические характеристики ВГТ-220-1К УХЛ1,4000 А, 40кА

Наименование параметра	Технические характеристики
1	2
Номинальное напряжение, кВ	220
Наибольшее рабочее напряжение, кВ	252
Номинальный ток, А	4000
Номинальный ток отключения, кА	40
Нормированное процентное содержание аperiodической составляющей, %, не более	40
Нормированные параметры сквозного тока короткого замыкания, кА:	
наибольший пик (ток электродинамической стойкости)	100
среднеквадратичное значение тока за время протекания (ток термической стойкости)	40
время протекания тока термической стойкости, с	3

Продолжение таблицы 2

1	2
Нормированные параметры тока включения, кА:	
наибольший пик	100
начальное действующее значение периодической составляющей	40
Коммутация емкостного тока:	
- номинальный ток коммутации ненагруженной воздушной линии, А	125
- номинальный ток коммутации кабельной линии, А	250
- номинальный ток коммутации одиночной батареи конденсаторов, А	400
- нормированный ток отключения шунтирующего реактора, А	315±63
Минимальный ток отключения шунтирующего реактора, А	100±20
Собственное время отключения, с	0,035 _{0,005}
Полное время отключения, с	0,055 _{-0,005}
Собственное время включения, с, не более	0,062 _{-0,018}
Разновременность работы полюсов, с, не более:	
при включении	0,005
при отключении	0,0033
Расход газа на утечки в год, % от массы газа, не более	0,5
Испытательное напряжение грозового импульса, к В относительно земли/ между разомкнутыми контактами	900/1050
Испытательное одномоментное напряжение промышленной частоты, кВ относительно земли/ между разомкнутыми контактами	440/440
Номинальное напряжение постоянного тока электромагнитов управления приводов, В	= 110, =220 или ~230
Количество электромагнитов управления:	
включающих	1
отключающих	2 или 3
Диапазон рабочих напряжений электромагнитов, % от номинального значения:	
включающий электромагнит	80-110
отключающий электромагнит	65-120

Выключатель состоит из трёх механически не связанных между собой полюсов колонкового типа, каждый из которых установлен на своей раме и управляется своим пружинным приводом типа ППрК-УЭТМ. Внешний вид выключателя представлен на рисунке 11.

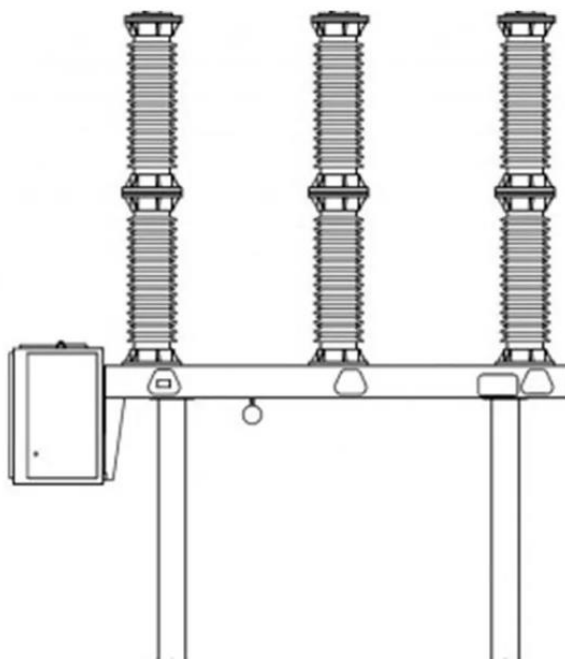


Рисунок 11 – Общий вид ВГТ-220-1К УХЛ1,4000 А, 40кА

Полюс выключателя состоит из двух колонн, гасительные устройства которых установлены на опорных изоляторах и соединены последовательно двумя шинами. Для равномерного распределения напряжения по гасительным устройствам параллельно к ним подключены шунтирующие конденсаторы.

Включение выключателей осуществляется за счёт энергии включающих пружин привода, а отключение - за счёт энергии пружин отключающего устройства.

Каждая колонна имеет свой электроконтактный сигнализатор плотности газа. Сигнализатор содержит устройство температурной компенсации, приводящее показания давления к температуре 20°C с тремя парами контактов, разомкнутых при номинальном (рабочем) давлении газа. Первая пара контактов замыкается и подаёт сигнал о необходимости пополнения колонны (давление предупредительной сигнализации). Вторая и третья пары контактов замыкаются и подают сигнал о необходимости включения блокировки подачи команды на электромагниты управления или сигнал принудительного отключения выключателя с запретом на его включение.

Количество разрывов электрической цепи - 2 разрыва на полюс.

Во всех соединениях используются двойные уплотнения, а в узле уплотнения подвижного вала – «жидкостный затвор».

Рассмотрим принцип работы выключателя. Для этого необходимо подробнее изучить дугогасительную камеру [5]. Состав камеры представлен на рисунке 12.



Рисунок 12 – Состав дугогасительной камеры

Дугогасительная камера сконструирована с учётом повышения механической и электрической прочности активной части и использования преимуществ незначительного износа контактов под воздействием дуги в элегазе. Активная часть заключена в герметичную керамическую оболочку, обеспечивающую развязку между входом и выходом автоматического выключателя.

Принцип отключения основан на гашении дуги дутьём с дополнительным автопневматическим эффектом [5].

В положении «закрото» ток проходит через соединитель (1), держатель неподвижного контакта (2), главные контакты (3), подвижный контакт (4), держатель подвижного контакта (5) и противоположный соединитель (1).

Электрическая или ручная команда отключения высвобождает энергию, накопленную в отключающей пружине (7), установленной в распределительном устройстве.

Изолирующий шунт (8), который приводится в движение непосредственно пружиной отключения (7), возбуждает подвижный контакт (4) и обеспечивает его размыкание. Таким образом, установленные контакты образуют электрическую дугу и погружаются в газовую среду. Дуга расщепляет газ на отдельные компоненты, но при этом и сама снижается из-за высокого давления в емкости. Кроме того, для балансировки тока используется шунт.

Управление выключателем осуществляется при помощи автоматики управления выключателем (АУВ), посредством протокола быстрой передачи данных о событиях на подстанции – GOOSE (Generic Object Oriented Substation Event) в переводе как «общее объектно-ориентированное событие на подстанции». Подробнее об АУВ рассказывается в пункте 2.2.1.

3.2.3 Выключатель на линии трансформатора

Выключатель на линии трансформатора отличается от выключателя на линии Юктали тем, что он другого вида, и имеет другой тип привода.

Управление выключателем осуществляется при помощи автоматики управления выключателем (АУВ), посредством протокола быстрой передачи данных о событиях на подстанции – GOOSE.

На рисунке 13 представлен элемент принципиальной схемы ОРУ 220 кВ ПС Хани. На котором изображена 6 Ячейка подстанции Хани 220 кВ.

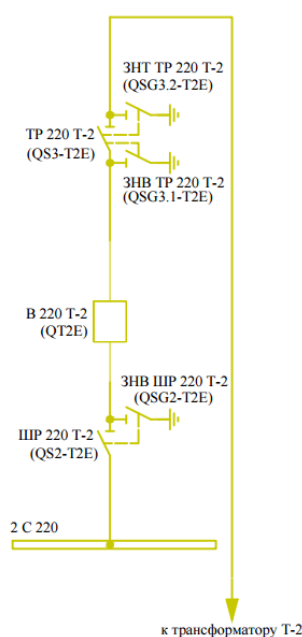


Рисунок 13 – Принципиальная схема 6 ячейки Т-2

Данным участком подстанции управляет Шкаф УСО №2. Схема подключения к его терминалу управления идентична со схемой подключения В 220 Юктали. От выключателя В 220 Т-2 приходит 2 сигнала о положении:

- 1) Положение выключателя – «Отключен»;
- 2) Положение выключателя – «Включен»;

Данный В 220 Т-2 типа HPL – 245 В1, 400А, 31,5 кА [4]. Внешний вид выключателя представлен на рисунке 14.

Его номинальное напряжение 72 – 800 кВ и ток отключения до 63 кА, а время для отключения тока КЗ в течение максимум 40 мс.

Выключатели HPL работают от приводов типа BLG со взводом пружин электродвигателями. Такой же тип привода установлен в секционном выключателе и принцип его работы был описан в пункте 3.2.3.

Выключатели HPL могут работать с 1-но и 3-х полюсным управлением. Выключатели, имеющие одно дугогасительное устройство на полюс, могут работать в обоих режимах управления. Выключатели с двумя дугогасительными устройствами на полюс допускают управление только на один полюс, т.е. пофазное.

Три полюса выключателя установлены на независимых полюсных опорах. При трехполюсной работе полюса выключателя и привода соединены тягами. Каждый полюс выключателя оснащен собственной пружиной отключения.

Каждый полюс выключателя собран в герметичную колонну заполненную элегазом, состоящую из дугогасительной камеры, полого опорного изолятора и корпуса механизма.

Поскольку герметичность зависит от плотности газа, выключатель HPL оснащен устройством контроля плотности газа. Монитор плотности представляет собой реле давления с температурной компенсацией, поэтому функция сигнализации и подавления срабатывает только в том случае, если утечка газа вызывает падение давления газа.

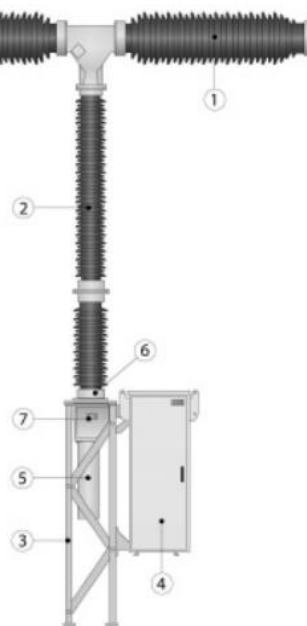


Рисунок 14 – Общий вид HPL – 245 В1, 400А, 31,5 кА

В состав HPL – 245 В1 входят [4]:

1. Дугогасящая камера;
2. Опорный изолятор;
3. Опорная конструкция;
4. Шкаф управления с приводом типа BLG;
5. Отключающая пружина;
6. Монитор плотности газа (с противоположной стороны);
7. Указатель положения выключателя;

Т.к. принцип работы всех элегазовых выключатель схож, то рассмотрим лишь принцип работы привода BLG.

Операция включения представлена на рисунке 15.

При включении выключателя катушка включения освобождает включающую защелку (6). Звездочка (7) запирается и не допускает вращения, вследствие чего энергия, запасенная во включающих пружинах, передается через секцию цепи (8) на звездочку (11), принадлежащую кулачку (3). Кулачок при этом перемещает рабочий рычаг (2) влево, где он фиксируется в конечном положении отключающей защелкой (1).

На конечном участке вращения кулачка амортизируется демпфирующим устройством (9), а фиксирующая защелка на звездочке (11) снова занимает исходное положение у включающей защелки (6).

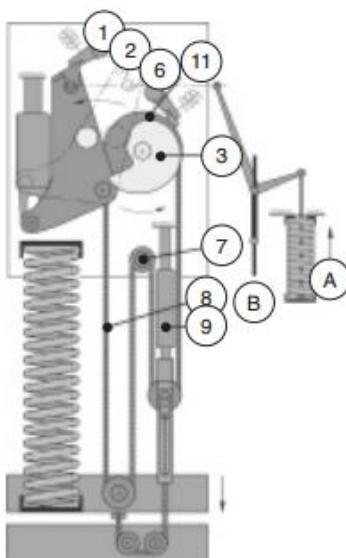


Рисунок 15 – Работа привода при включении выключателя

Операция отключения представлена на рисунке 16.

При отключении выключателя катушка отключения освобождает защелку (1).

Отключающая пружина (А) выполняет операцию отключения выключателя (В).

Рабочий рычаг (2) перемещается вправо и упирается в кулачок (3). Движение системы вспомогательных контактов амортизируется в конце хода масляным демпфирующим устройством (4).

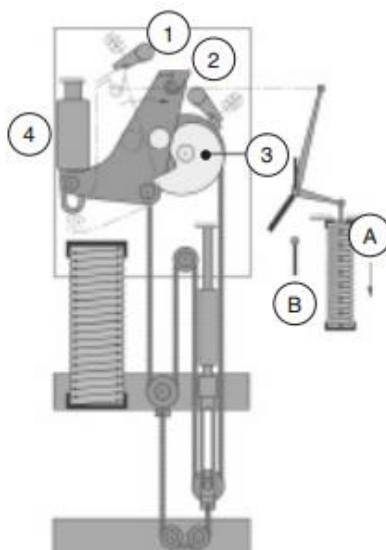


Рисунок 16 – Работа привода при отключении выключателя

После разбора работы привода необходимо разобрать его принципиальную электрическую схему. Принципиальная электрическая схема представлена на рисунке 17.

Цепь включения - катушка включения (Y3) может быть включена электрически с места ключом, из шкафа привода или дистанционно со щита управления. Когда выключатель находится в положение «включено», цепь включения размыкается вспомогательным контактом (BG).

Цепи отключения - привод имеет две независимые катушки отключения (Y1 и Y2). Приводом можно управлять электрически с места ключом из шкафа привода или дистанционно со щита управления. Когда выключатель находится в положение «отключено», цепи отключения размыкаются вспомогательным контактом (BG).

Блокировки - контакт монитора плотности (BD) выдает сигнал вспомогательным реле (K9, K10), которые блокируют управляющий импульс если плотность элегаза слишком мала. Реле блокировки от многократных включений (K3) блокирует любые последующие импульсы на включение после выполнения выключателем операции включения.

В зависимости от плотности элегаза и состояния привода выдаются следующие выходные сигналы:

- 1) Рекомендуется дозакачка элегаза (аварийный уровень);
- 2) Слишком низкая плотность элегаза (уровень блокировки);
- 3) Пускатель для прямого пуска электродвигателя отключен от сети;
- 4) Низкая энергия пружины.

Цепи нагревателей - привод оборудован антиконденсационным нагревателем. Чтобы обеспечить надежную работу при низких температурах привод оборудован нагревательным блоком (E2), управляемым термостатом (BT1). В другом исполнении привод может быть оборудован датчиком влажности для эксплуатации в климате с высокой влажностью [4]. Полная принципиальная электрическая схема представлена в приложении В.

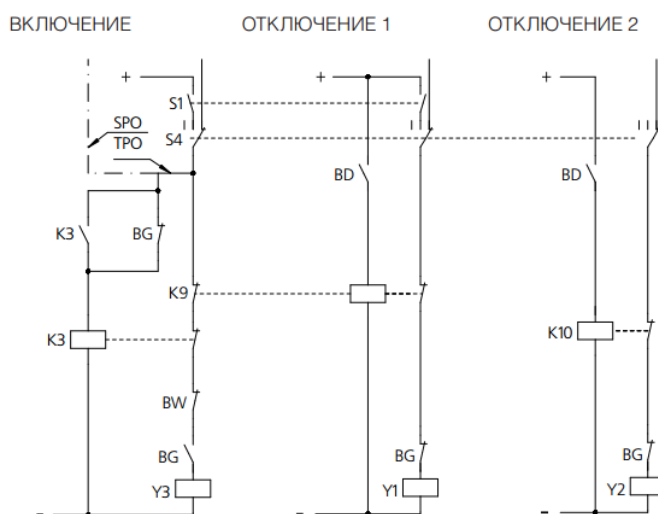


Рисунок 17 – Принципиальная электрическая схема привода BLG

BD - Сигнальный контакт монитора плотности;

BG - Вспомогательный контакт;

BT1 – Термостат;

BW – Конечный выключатель;

E1, E2 – Нагреватель;

F1, F1.1 – Пускатель для прямого пуска электродвигателя от сети (микровыключатель);

F2 – Микровыключатель, вспомогательные цепи переменного тока;

K3 – Реле блокировки от многократных включений;

К9, К10 – Вспомогательное реле, блокировка при низкой плотности эле-
газа, отключение, включение;

М, М.1 – Электродвигатель;

Q1, Q1.1 – Контактор;

S1 - Ключ управления (Отключение / Включение);

S4 – Многопозиционный переключатель выбора режима управления
(«Местное / Дистанционное / Отключено»);

Y1, Y2 – Катушка отключения;

Y3 – Катушка включения;

Y7 – Блокировочный контакт (при ручном взводе пружин).

4 РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ПРОГРАММЫ ПЕРЕ- КЛЮЧЕНИЙ

Данная автоматизированная программа переключений (далее - АПП) предназначена для управления основными коммутирующими аппаратами ОРУ 220 кВ на подстанции Хани.

В разделе 3 уже говорилось, что все коммутирующие аппараты оснащены собственными приводами, которые допускают как местное, так и дистанционное управление. В последнем случае команды управления в виде дискретных сигналов поступают от специализированного контроллера - Advantech ECU-4784-D5SBE, соединенного посредством тех или иных каналов связи с оборудованием диспетчерского пункта. В процессе управления контроллер получает информацию о состоянии аппарата и режиме его функционирования. Эта информация, которая также носит дискретный характер, далее передается диспетчеру.

Важным обстоятельством является то, что по условиям безопасности аппараты требуют согласованного управления. В частности, запрещается проводить операции с разъединителем под напряжением (при включенном выключателе), опускать заземляющий нож при включенном разъединителе и т.д. Соответствующие блокировки реализованы как на уровне электрических схем управления приводами аппаратов, так и в программе управляющего контроллера. Таким образом, действия диспетчера, манипулирующего элементами управления на своем экране визуализации, не могут привести к фатальным последствиям.

В данной работе предложен подход к решению этой проблемы, основанный на использовании технологии объектно-ориентированного программирования, которая все более активно применяется в промышленной автоматизации.

Одним из лидеров продвижения объектно-ориентированного подхода в практику программирования ПЛК является немецкая компания CODESYS GmbH (старое название 3S-Smart Software Solutions GmbH), входящая в группу компаний CODESYS Group [6]. Ее SoftLogic и runtime систему CODESYS (Controller Development System) используют многие производители ПЛК по

всему миру, в том числе российская компания ОВЕН. В последних версиях системы в качестве расширения стандарта IEC 61131-3 реализована поддержка практически всех основных элементов объектно-ориентированного программирования.

Автоматизированная программа переключений состоит из двух составляющих или подсистем:

- 1) Подсистема формирования и хранения режима переключений.
- 2) Подсистема выполнения плана переключений.

4.1 Подсистема формирования и хранения режима переключений

Данная подсистема формирует шаблон, содержащий всю необходимую информацию для работы подсистемы выполнения плана переключений.

Данную подсистему формируют два основных интерфейса:

- 1) Интерфейс формирования режима переключений;
- 2) Интерфейс управления.

Так как данная подсистема состоит из интерфейсов, следовательно именно с ней и будет взаимодействовать оперативный работник или инженер АСУ. Поэтому данная подсистема сформирована таким образом, чтоб в ходе эксплуатации пользователь данной АПП не смог создать ЧС на подстанции.

4.1.1 Интерфейсы подсистемы

Сперва разберём интерфейс формирования режима переключений. Данный интерфейс, это первое что увидит пользователь программы. Интерфейс представлен на рисунке 18.

Именно в этом интерфейсе формируется:

- 1) Выбор КА;
- 2) Команда для КА (включение/отключение);
- 3) Последовательность выполнения команд.

После выбора необходимых КА и задания последовательности переключения, программу можно будет запускать в работу. Но также вышеперечисленные действия формируют так называемый шаблон. Под шаблоном подразумевается заполненный интерфейс программы, со всеми выбранными КА их ячейками и

операциями над ними. На рисунке 18 демонстрируется заранее созданный шаблон для включения в работу 7 ячейки подстанции

Шаблон содержит в себе:

- 1) Текстовые поля с выбранными ячейками подстанции;
- 2) Текстовые поля с КА выбранной ячейки;
- 3) Текстовые поля с операцией (включить/отключить) над КА;
- 4) Текстовые поля с датой и временем переключения выбранных КА.

Также после всего вышеперечисленного данная подсистема может сохранять созданные пользователем шаблоны при нажатии кнопки «Записать в файл», и загружать ранее сохранённые шаблоны при нажатии кнопки «Прочитать из файла».

Ячейка	Объект	Операция	Дата и время	
ВЛ 220кВ Юктали-Хани	ШР 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.
ВЛ 220кВ Юктали-Хани	ЛР 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.
ВЛ 220кВ Юктали-Хани	ШР 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.
ВЛ 220кВ Юктали-Хани	В 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.

Информация:

Добавить задание Удалить задание Записать в файл Прочитать из файла

Выполнить

Рисунок 18 – Интерфейс формирования режима переключений

Условия и порядок действий для создания АПП:

- 1) Нужно выбрать ячейку подстанции с необходимым КА;
- 2) Выбрать объект переключения, т.е. КА;
- 3) Затем выбрать операцию для КА (включение/отключение);
- 4) После нажать на кнопку «Актив.». Она записывает выбранные перечисленные выше условия в строки под выбором действий. Каждая строка имеет

свою определённую переменную типа – STRING. После необходимо выполнить те же действия для другого КА. Заполнять строки необходимо в чёткой последовательности действий над КА. Именно от этих строк зависит подсистема выполнения плана переключений;

5) Затем необходимо нажать кнопку «Выполнить». Она изменяет показанную визуализацию, т.е. перебрасывает пользователя в интерфейс управления. Данный интерфейс представлен на рисунке 19.

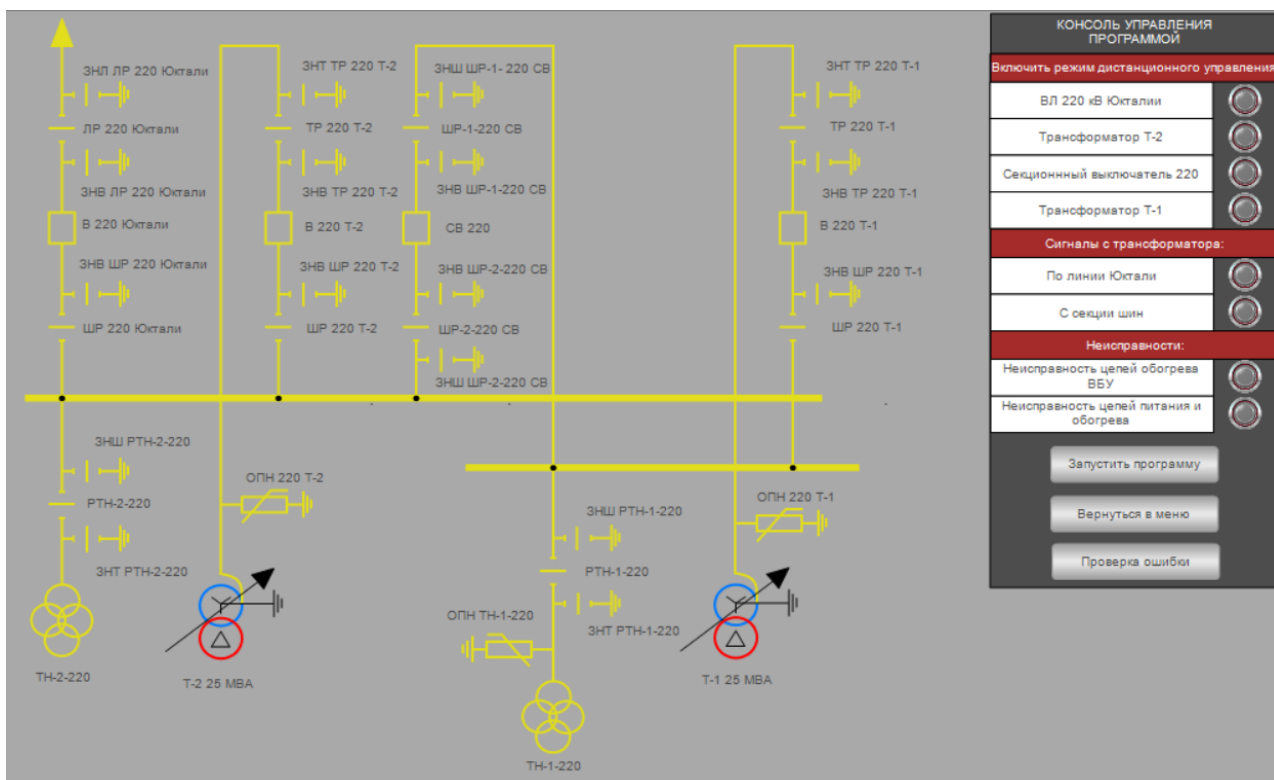


Рисунок 19 – Интерфейс управления

В интерфейсе управления происходит запуск программы на выполнение. Также в данный интерфейс добавлены сигналы имитирующие неисправности в шкафу привода определённого КА, и сигналы с трансформаторов. Сделано это для испытания программы вне подстанции.

Интерфейс снабжён схемой имитирующую схему SCADA-системы. На данной схеме отображается работа программы, т.е. переключение КА.

Также в интерфейсе присутствует консоль управления программой. Данная консоль необходима для задания подтверждающих сигналов, необходимых

для удовлетворения условий переключения. Консоль управления включает в себя несколько разделов:

1) «Дистанционный режим». В данном разделе происходит переключение режима управления КА. В разделе 3 уже говорилось о внутреннем блоке управления (далее ВБУ) привода, он выбирает режим работы привода (местное/дистанционное), чаще всего привод находится в дистанционном режиме управления, но также ВБУ находится и в шкафу УСО. Он также выбирает режим работы привода, но не одного, а приводов всей ячейки, которые относятся к шкафу УСО. Кнопки в данном разделе и управляют устройствами ВБУ в шкафах УСО.

2) «Сигналы с трансформаторов». Как говорилось выше, данный раздел предназначен для имитации сигналов с трансформаторов, а именно с трансформатора по приходящей ВЛ Юктали и с секционного трансформатора.

3) «Неисправности». Данный раздел также как и предыдущий предназначен для имитации сигналов. Неисправности указаны в подразделе - 3.2, на схемах подключения к терминалу управления ЭКРА 242.

При нажатии на любую из кнопок раздела появится диалоговое окно, в котором будут указаны КА, имеющие данный сигнал неисправности. Диалоговое окно неисправностей цепей питания и обогрева представлено на рисунке 20.



Рисунок 20 – Диалоговое окно неисправностей цепей питания и обогрева

Также на консоли присутствуют кнопки управления программой:

Кнопка «Запустить программу». Данная кнопка запускает программу на исполнение.

Кнопка «Вернуться в меню». Данная кнопка переносит пользователя в Интерфейс формирования режима переключений.

Кнопка «Проверка ошибки». Данная кнопка выводит диалоговое окно, в котором будет указана ошибка переключения. Также данное окно откроется автоматически в случае ошибки переключений. Окно представлено на рисунке 21.



Рисунок 21 – Окно ошибок

4.1.2 Программный код подсистемы

Весь программный код данной подсистемы записан в различные кнопки, установленные на интерфейсах, поэтому будем рассматривать код, ссылаясь на определённые кнопки.

Разберём кнопки относящиеся к интерфейсу формирования режима переключений.

Кнопка «Актив.» - Программный код кнопки показан ниже.

```
IF cTask>0 THEN
    Card.Tasks[cTask].Cell:=Cells[nCell];
    Card.Tasks[cTask].Object:=Objects[nObject];
    Card.Tasks[cTask].WorkType:=WorkTypes[nWorkType];
    Card.Tasks[cTask].DateAndTime:=DateAndTime;
END_IF
```

Данный код, записывает в структуру Task (подробнее структура описана в подразделе 4.2) данные, выбранные в выпадающих списках, каждый выпадающий список имеет свой массив данных, если выполнено верхнее условие.

Выпадающий список ячеек содержит массив данных Cells и его переменную nCell.

Выпадающий список объектов (т.е. коммутирующих аппаратов) содержит массив данных Objects и его переменную nObject.

Выпадающий список операций содержит массив данных WorkTypes и его переменную nWorkType.

Также в структуру записывается переменная, содержащая дату и время переключения – DateAndTime.

Все вышеперечисленные переменные записываются в структуру Task и переменные данной структуры отображаются в текстовых полях под выпадающими списками, образуя строку как показано на рисунке 18.

Так как кнопок «Актив.» всего 6 штук, то код показан для первой кнопки. У последующих кнопок различается лишь одна часть кода – “сTask:=1;”. В зависимости от порядка расположения кнопки (сверху вниз) число меняется и данные кнопки перезаписывают структуру и меняют отображаемые данные строки текстовых полей, находящихся напротив кнопки «Актив.».

Кнопка «Добавить задание» - Программный код кнопки показан ниже.

```
IF Card.nTask < 5 THEN
    Card.nTask:=Card.nTask+1;
    сTask:=Card.nTask;
END_IF
```

Данный код увеличивает переменную nTask на 1 единицу и записывает значение данной переменной в переменную сTask тем самым, позволяя выполнять код кнопки «Актив.». Увеличивая переменную сTask увеличивается кол-во видимых строк пользователя. Что позволяет добавить больше КА, и создать последовательность переключений определённых КА.

Кнопка «Удалить задание» - Программный код кнопки показан ниже.

```
Card.nTask:=Card.nTask-1;
```

```

IF cTask > Card.nTask THEN
    cTask:=Card.nTask;
END_IF

```

Данный код противоположен коду кнопки «Добавить задание». Он уменьшает переменную nTask на 1 единицу и если cTask больше nTask, то значение nTask записывается в переменную cTask. Уменьшая переменную cTask уменьшается кол-во видимых строк пользователя.

Кнопка «Записать в файл» - Программный код кнопки показан ниже.

Код для конфигурации ввода OnDialogClosed:

```

IF updating THEN
    stDir:=fbdFLP.GetSelectedDirectory();
    stFile:=fbdFLP.GetSelectedFileName();
    fullPath:=CONCAT(stDir,stFile);
    FileD:=VisuDialogs.SysFile.SysFileOpen(fullPath,
        VisuDialogs.SysFile.AM_WRITE, ADR(rte));
    VisuDialogs.SysFile.SysFileWrite(FileD,ADR(Card),
        SIZEOF(Card),ADR(rte));
    rte:=VisuDialogs.SysFile.SysFileClose(FileD);
END_IF
updating:=FALSE;

```

Код для конфигурации ввода OnMouseClicked:

```

fbdFLP.Initialize
(
bBrowseDirectory:=TRUE,
bTouch:=FALSE,
iRowCount:=2,
stDirectory:=' ',
stFileIn:='card.crd',
stFilter:='All files (*.crd)|*.crd',
stTitle:='Select file'
);
updating:=TRUE;

```

Данный код позволяет записывать шаблон АПП в файл и сохранять его для дальнейшего хранения и использования в будущем. Он создаёт файл и вызывает диалоговое окно, в котором выбирается место сохранения файла с шаблоном АПП.

Работа с файлами осуществляется благодаря встроенной в CODESYS 3.5. библиотеке VisuDialogs и встроенной в неё библиотеке SysFile.

При нажатии на кнопку сначала происходит инициализация функционального блока `fbdFLP`, данный ФБ является встроенным в библиотеку `VisuDialogs`, здесь задаются необходимые параметры, такие как тип файла, фильтр для поиска и директория. Затем меняется значение переменной `updating`, что позволяет сохранить файл в необходимом для пользователя месте.

Кнопка «Прочитать из файла» - Программный код кнопки показан ниже.

Код для конфигурации ввода `OnDialogClosed`:

```
IF updating THEN
    stDir:=fbdFLP.GetSelectedDirectory();
    stFile:=fbdFLP.GetSelectedFileName();
    fullPath:=CONCAT(stDir,stFile);
    FileD:=VisuDialogs.SysFile.SysFileOpen(fullPath,
        VisuDialogs.SysFile.AM_READ, ADR(rte));
    VisuDialogs.SysFile.SysFileRead(FileD,ADR(Card),
        SIZEOF(Card),ADR(rte));
    rte:=VisuDialogs.SysFile.SysFileClose(FileD);
END_IF
updating:=FALSE;
```

Код для конфигурации ввода `OnMouseClicked`:

```
fbdFLP.Initialize
(
bBrowseDirectory:=TRUE,
bTouch:=FALSE,
iRowCount:=2,
stDirectory:='./',
stFileIn:='card.crd',
stFilter:='All files (*.crd)|*.crd',
stTitle:='Select file'
);
updating:=TRUE;
```

Данный код позволяет загрузить необходимый шаблон АПП, сохранённый в файле. Для этого в программе вызывается диалоговое окно, в котором выбирается необходимый сохранённый файл с шаблоном АПП.

Работа с файлами осуществляется благодаря встроенной в CODESYS 3.5. библиотеке `VisuDialogs` и встроенной в неё библиотеке `SysFile`.

При нажатии на кнопку сначала происходит инициализация функционального блока `fbdFLP`, данный ФБ является встроенным в библиотеку

VisuDialogs, здесь задаются необходимые параметры, такие как тип файла, фильтр для поиска и директория. Затем меняется значение переменной `updating`, что позволяет прочитать сохранённый шаблон АПП.

Теперь разберём кнопки относящиеся к интерфейсу управления.

Кнопки имеющие свой программный код содержатся в разделе «Включить режим дистанционного управления», код данных переключателей схож, поэтому достаточно разобрать лишь программный код одного переключателя.

Переключатель режима управления «ВЛ 220 кВ Юктали»:

Программный код переключателя показан ниже.

```
GVL.Isp_Mech[1].Work_Mode:=TRUE;  
GVL.Isp_Mech[4].Work_Mode:=TRUE;  
GVL.Isp_Mech[8].Work_Mode:=TRUE;  
GVL.Isp_Mech[13].Work_Mode:=TRUE;  
GVL.Isp_Mech[17].Work_Mode:=TRUE;  
GVL.Isp_Mech[18].Work_Mode:=TRUE;  
GVL.Isp_Mech[19].Work_Mode:=TRUE;  
GVL.Isp_Mech[21].Work_Mode:=TRUE;  
GVL.Isp_Mech[31].Work_Mode:=TRUE;
```

Данный код меняет режима управления ВБУ находящегося в приводах КА относящихся к ячейке - «ВЛ 200 кВ Юктали». Т.е. данный переключатель является ВБУ стоящем в шкафу УСО 1 ячейки 7.

В данном коде приводом является структура `Isp_Mech`, а число в квадратных скобках указывает на КА к которому относится привод. Переменная `Work_Mode` содержащаяся в структуре, является ВБУ привода и меняется в зависимости от нажатия на переключатель.

Остальные переключатели режимов управления, содержат в себе идентичный код, но каждый имеет свой набор КА относящихся к своей ячейке подстанции.

Остальные кнопки программы содержат лишь переменные типа `BOOL` и отвечают лишь за переключение привязанных к ним переменных. Рассматривать их не имеет смысла.

4.2 Подсистема выполнения плана переключений

Данная подсистема необходима для выполнения плана переключений. Она считывает шаблон, созданный подсистемой формирования и хранения режима переключений, и исходя из данных в шаблоне она находит нужный КА в своей «базе данных», и затем выполняет необходимую операцию над ним, указанную в шаблоне.

Вся подсистема является программным кодом программы. Весь программный код подсистемы состоит из функциональных блоков, компонентов организации программ (далее - РОУ) и структур.

4.2.1 Структуры программы

Для начала рассмотрим структуры программы, ведь они фигурируют в РОУ и в ФБ.

Что же такое структура? Структура – это определяемый пользователем тип данных, который объединяет несколько переменных любого типа данных в логическую единицу.

Имя структуры – это тип данных, который CODESYS распознаёт во всем проекте. Обращение к компонентам структуры выполняется с использованием следующего синтаксиса: «Имя структуры». «имя компонента». Использование данного синтаксиса показано в приложении Ж.

Структура «Dialog» - инициализация структуры показана ниже.

```
TYPE Dialog :  
STRUCT  
    Name:      STRING;  
    Message:   STRING;  
END_STRUCT  
END_TYPE
```

Данная структура предназначена для диалогового окна ошибок, изображённого на рисунке 21, в пункте 4.1.1.

Структура содержит: компонент Name указывающий на имя КА в котором произошла ошибка, а также компонент Message содержащий сообщение о причине ошибки у КА. Оба этих компонента демонстрируются в текстовых полях диалогового окна.

Структура «I_Mech» - инициализация структуры показана ниже.

```
TYPE I_Mech :  
STRUCT  
    Work_Mode:      BOOL;  
    ON:             BOOL:=FALSE;  
    OFF:           BOOL:=TRUE;  
    position:      REAL:=0;  
    integ:         INTEGRAL;  
END_STRUCT  
END_TYPE
```

Данная структура содержит в качестве компонентов сигналы с привода КА, а также дополнительные переменные, для POU IM.

Компонент `Work_Mode` является сигналом о режиме работы привода (местное/дистанционное). Т.е. он является ВБУ самого привода. Если данная переменная имеет значение `FALSE`, это означает что привод находится в МЕСТНОМ режиме работы, если переменная имеет значение `TRUE`, то привод находится в ДИСТАНЦИОННОМ режиме работы.

Компонент `ON` является сигналом об включенном положении привода. Поэтому изначально он имеет значение `FALSE`, так как привод отключен. Если значение изменится на `TRUE`, это означает что привод перешёл в положение включено и включил КА независимо от того будь это разъединитель или выключатель.

Компонент `OFF` является сигналом об отключенном положении привода. Поэтому изначально он имеет значение `TRUE`, так как привод отключен. Если значение изменится на `FALSE`, это означает что привод перешёл в положение включено и включил КА независимо от того будь это разъединитель или выключатель.

Компонент `position` является переменной, обозначающей положение привода в интервале от 0.5 до 99.5. Если `position` меньше 0.5 это отключенное положение, а если `position` больше 99.5 это включенное положение привода. Промежуточные значения являются ошибочными и сигнализируют о неисправности привода или всего КА.

Компонент `integ` является функциональным блоком `INTEGRAL`. Данный функциональный блок будет аппроксимировать интегральную функцию функции ($f = f(t)$) за интервал времени между первым вызовом функции и фактическим временем. Данный ФБ необходим для более точного имитирования работы привода, именно этот блок изменяет компонент `position` в определённом временном интервале.

Структура «Task» - инициализация структуры показана ниже.

```
TYPE Task :
STRUCT
    Cell:          STRING(30);
    Object:        STRING(20);
    WorkType:      STRING(20);
    DateAndTime:  DT;
END_STRUCT
END_TYPE
```

Данная структура необходима для работы подсистемы формирования и хранения режима переключений. Ранее структура упоминалась в пункте 4.1.2. В компоненты структуры записываются данные, выбранные в выпадающих списках: «Ячейка», «Объект» и «Операция», а также данные из текстового поля «Дата и Время» переключения КА.

В компонент `Cell` записываются данные из выпадающего списка «Ячейка». В компонент `Object` записываются данные из выпадающего списка «Объект». В компонент `WorkType` записываются данные из выпадающего списка «Операция». В компонент `DataAndTime` записывается информация из текстового поля «Дата и Время».

Структура «WorkCard» - инициализация структуры показана ниже.

```
TYPE WorkCard :
STRUCT
    Title:          STRING(50);
    Tasks:          ARRAY[1..6] OF Task;
    nTask:          BYTE;
    Information:    STRING(100);
END_STRUCT
END_TYPE
```


Данная структура необходимо для полноценной работы программы, она используется во всех подсистемах программы и необходима для корректной работы с файлами. Именно она записывается в файл в виде переменной `Card`. Данная переменная указана в приложении Ж, в разделе глобальных переменных.

Компонент `Title` является переменной, в которую записываются данные из текстового поля «Заголовок».

Компонент `Tasks` является массивом структур `Task`, это означает, что переменная `Tasks` содержит в себе шесть структур `Task`. Это необходимо для правильного формирования строк в шаблоне АПП. Каждой строке соответствует одна заполненная структура `Task`.

Компонент `nTask` это номер структуры. Он необходим для выбора определённой структуры `Task` в массиве структур `Tasks`.

Компонент `Information` является переменной, в которую записываются данные из текстового поля «Информация».

4.2.2 Функциональные блоки программы

Для создания функциональных блоков был реализован виртуальный базовый класс. Виртуальные базовые классы в CODESYS 3.5 называется интерфейсами. Интерфейсы декларируют свойства и методы, которые должны реализовать наследующие от них классы (функциональные блоки). Причем в отличие от C++ в CODESYS разделяются понятия «обычного» наследования (`extending` – расширение) и наследования от виртуального класса (`implementation` – реализация). Организационная структура данных показана на рисунке 22.

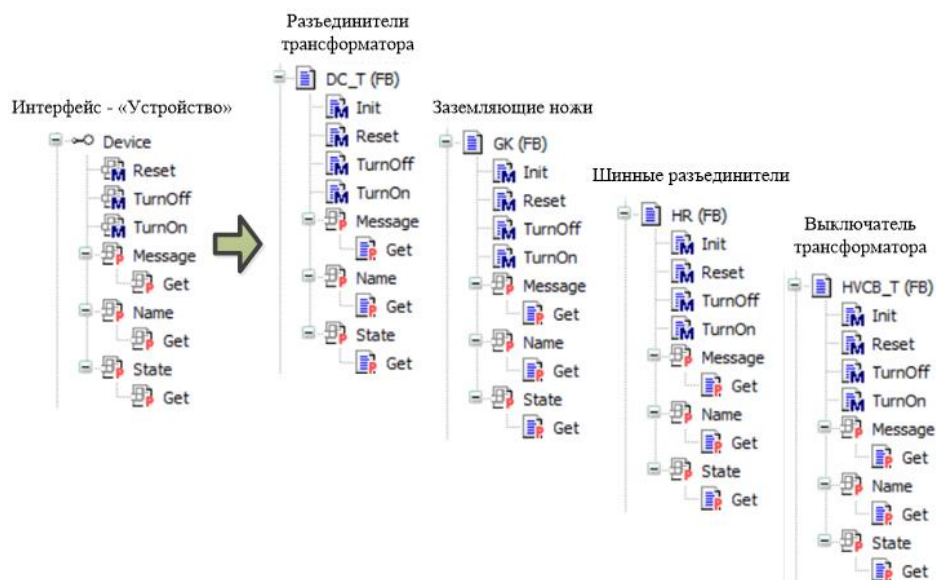


Рисунок 22 – Интерфейс и функциональные блоки

Интерфейс Device («устройство»), реализуемый всеми функциональными блоками программы: DC_LR (линейный разъединитель на линии Юктали), DC_T (разъединители на линиях понижающих трансформаторов), GK (Заземляющие ножи разъединителей), GKH (заземляющие ножи для секций шин), HR (шинные разъединители), HVCB_SV (секционный выключатель), HVCB_T (выключатели на линиях понижающих трансформаторов), HVCB_UKT (выключатель на линии Юктали), RTN (разъединитель измерительного трансформатора тока) включает:

- 1) метод Reset (сброс после ошибки);
- 2) методы TurnOn и TurnOff (включить/выключить аппарат);
- 3) свойство Message (последнее сообщение от аппарата);
- 4) свойство Name (идентификатор аппарата);
- 5) свойство State (состояние аппарата).

Для всех свойств реализации подлежат методы Get (прочитать значение), но не Set (установить).

В функциональные блоки добавлены методы Init, предназначенные для инициализации экземпляров. В CODESYS можно использовать и настоящие *конструкторы* – методы FB_Init, вызываемые при объявлении экземпляров, однако необходимость в наличии перекрестных ссылок мешает их применению.

Для решения своих задач экземпляры должны знать своих соседей, поэтому они сначала создаются (объявляются), а потом инициализируются.

Все экземпляры функциональных блоков объявляются в глобальном списке и инициализируются специальной программой Initialization, которая вызывается однажды при старте системы.

Функциональный блок «DC_LR» - данный функциональный блок (далее - ФБ) отвечает за работу ЛР 220 Юктали. Это значит, что все его методы и сам ФБ запрограммированы на получение сигналов и их обработку только с ЛР 220 Юктали. Подключить к данному ФБ заземляющий нож или шинный разъединитель невозможно.

Функциональные блоки содержат полный набор данных, необходимых для их работы. Ниже приведен список локальных переменных функционального блока DC_LR.

```
FUNCTION_BLOCK DC_LR IMPLEMENTS Device
VAR_INPUT
  inline:BOOL;
END_VAR
VAR_OUTPUT
  outline:BOOL;
END_VAR
VAR
  myName:          STRING;
  myState:         STRING:='off';
  myMessage:      STRING:='';
  //указатель на выключатель В 220 Юктали
  myHVCB_UKT:     POINTER TO HVCB_UKT;
  //указатель на ЗНЛ ЛР 220 Юктали
  myGKL:          POINTER TO GK;
  //указатель на ЗНВ ЛР 220 Юктали
  myGKV:          POINTER TO GK;
  //указатель отображения работы ФБ
  myActiv:        POINTER TO BOOL;
  //указатель на вход - состояние аппарата (включен)
  myDet_vkl:      POINTER TO BOOL;
  //указатель на вход - состояние аппарата (отключен)
  myDet_otkl:     POINTER TO BOOL;
  //указатель на измерительный ТТ
  myTransformator: POINTER TO BOOL;
  //указатель на неисправность цепей питания и обогрева
```

```

myDefect_circuit:  POINTER TO BOOL;
//указатель на неисправность цепей обогрева ВБУ
myDefect_VBU:     POINTER TO BOOL;
timer:           TON; //таймер для контроля времени операций
END_VAR

```

Строковые переменные напрямую транслируются в соответствующие свойства их методами Get. Блок содержит ссылки на «соседей»: выключатель и заземляющие ножи. Оба объекта представлены как указатели на тип функционального блока, который может быть привязан к ФБ DC_LR, что позволяет коду блока читать их свойства и даже вызывать методы (последнее не используется). Связь с аппаратурой организуется посредством указателей на входы и выходы ПЛК.

Задачу контроля состояния аппаратов решает программная часть функциональных блоков. Она построена по автоматному принципу: в зависимости от состояния, в котором находится аппарат («включен», «выключен», «включается», «выключается», «ошибка» и др.), выполняются те или иные действия, сопровождаемые выдачей в свойство Message соответствующих сообщений. При этом, помимо собственного, контролируется также и состояние аппаратов-соседей. При достижении определенных условий производится смена состояния аппарата.

Ниже показана часть программного кода ФБ, отвечающая за включение разъединителя в работу. Полный код программы указан в приложении Ж.

```

ELSIF myState = 'GoOn' THEN
  IF myHVCB_UKT^.State='off' AND myGKL^.State='off' AND
myGKV^.State='off' AND NOT myTransformator^ AND NOT myDefect_VBU^
AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_vkl^ THEN
      myState:='on';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=TRUE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  END_IF

```

END_IF

Данная часть кода отслеживает состояние КА при его включении в работу. При `myState = 'GoOn'`, это означает что КА находится в состоянии «включение», проверяются условия для безопасного включения разъединителя, такие как состояния «соседей» КА, сигналы об неисправностях и сигнал с измерительного ТТ. Если все условия выполнены, происходит запуск таймера контроля времени операций. После, с привода КА приходит сигнал о включенном состоянии и состояние КА переходит с `GoOn` в `On`, таймер отключается и приходит сигналы о завершении работы КА. При истечении таймера, состояние КА переходит с `GoOn` в `fault`. Это означает что сигнал с привода не пришёл, следовательно привод вышел из строя и дальнейшая работа невозможна.

Часть кода, отслеживающая состояние КА при его отключении организована подобным образом, за исключением условий для отключения. Полный код функционального блока указан в приложении Ж.

В отличие от основной программы функционального блока, пересчитываемой циклически при циклическом его вызове, методы, являясь функциями, вызываются однократно и должны полностью выполнить свою задачу за один вызов. Поэтому методы «TurnOn» и «TurnOff» выполняют задачу управления приводами.

Разберём метод «TurnOn». Он необходим для подачи сигнала о том, что КА готов ко включению, если соблюдены все условия для его включения и переводит его из состояния «выключен» в состояние «включается», формирует соответствующее сообщение и возвращает `TRUE`, а также посылает сигнал включения, компоненту организации программы – `POU IM` (исполнительный механизм). Вся дальнейшая работа возлагается на основную программу ФБ. Если в данный момент включить аппарат не представляется возможным, формируется сообщение об этом и возвращается `FALSE`. Программный код метода указан в приложении Ж.

Методы «TurnOff» необходим для подачи сигнала о том, что КА готов к отключению, если соблюдены все условия для его отключения и переводит его

из состояния «включен» в состояние «отключается», формирует соответствующее сообщение и возвращает TRUE, а также посылает сигнал отключения, компоненту организации программы – POU IM (исполнительный механизм). Вся дальнейшая работа возлагается на основную программу ФБ. Если в данный момент включить аппарат не представляется возможным, формируется сообщение об этом и возвращается FALSE. Программный код метода указан в приложении Ж.

Метод «Reset» необходим для сброса после ошибки. При срабатывании данного метода, разъединитель будет переведён в состояние готовности к отключению. Программный код метода указан в приложении Ж.

Метод «Init». Он необходим для инициализации ФБ. Именно в нём задаются такие параметры как: идентификатор аппарата, положение привода, состояние аппарата, задаётся зависимый КА, а также сигналы о неисправностях. Программный код метода указан ниже.

Локальные переменные метода:

```
METHOD Init
VAR_INPUT
Name:                STRING;
pHVCB_UKT:           POINTER TO HVCB_UKT;
pGKL:                POINTER TO GK;
pGKV:                POINTER TO GK;
pActiv:              POINTER TO BOOL;
pDet_vkl:            POINTER TO BOOL;
pDet_otkl:           POINTER TO BOOL;
pTransformator:     POINTER TO BOOL;
pDefect_circuit:    POINTER TO BOOL;
pDefect_VBU:        POINTER TO BOOL;
END_VAR
```

Реализация метода:

```
myName:=Name;
myHVCB_UKT:=pHVCB_UKT;
myGKL:=pGKL;
myGKV:=pGKV;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myTransformator:=pTransformator;
myDefect_circuit:=pDefect_circuit;
```

```
myDefect_VBU:=pDefect_VBU;
```

Локальные переменные метода записываются в переменные ФБ. Ранее эти переменные были заданы в POU «Initialization».

Функциональный блок «HR» - данный функциональный блок отвечает за работу всех шинных разъединителей, указанных в программе. Это значит, что все его методы и сам ФБ запрограммированы на получение сигналов и их обработку только с шинных разъединителей, таких как ШР 220 Юктали. Подключить к данному ФБ заземляющий нож или выключатель невозможно.

Также как и ФБ DC_LR, HR содержит полный набор данных, необходимых для работы. Ниже приведен список локальных переменных функционального блока HR.

```
FUNCTION_BLOCK HR IMPLEMENTS Device
VAR_INPUT
  inline:BOOL;
END_VAR
VAR_OUTPUT
  outline:BOOL;
END_VAR
VAR
  myName:          STRING;
  myState:         STRING:='off';
  myMessage:      STRING:='';
  myHVCB:         Device; //ссылка на выключатель
  myGK:           POINTER TO GK;
  //указатель на ЗНШ ШР-2-220 СВ
  myGKH_1:        POINTER TO GKH;
  //указатель на ЗНШ РТН-2-220
  myGKH_2:        POINTER TO GKH;
  myActiv:        POINTER TO BOOL;
  myDet_vkl:      POINTER TO BOOL;
  myDet_otkl:     POINTER TO BOOL;
  myDefect_circuit: POINTER TO BOOL;
  myDefect_VBU:   POINTER TO BOOL;
  timer:TON;
END_VAR
```

В отличие от DC_LR, HR имеет вместо указателя на выключатель, ссылку на выключатель, так как ссылка может содержать любой тип ФБ программы, в отличие от указателя, но это также позволяет коду блока читать их свойства и даже вызывать методы.

Также ФБ HR имеет другие привязанные КА, ведь для его условий включения и отключения, необходимы КА которые влияют на работу шинных разъединителей, это выключатель и заземляющие ножи, такие КА как разъединители не влияют на работу шинного разъединителя, а значит и в программной реализации указывать их не имеет смысла.

В остальном функциональный блок отличается лишь условиями включения и отключения, так как имеет разные привязанные КА.

Методы и их программный код данного ФБ идентичны ФБ DC_LR, за исключением условий включения и отключения. Полный код ФБ и его методов указан в приложении Ж.

Функциональный блок «HVCB_T» - данный функциональный блок отвечает за работу выключателей установленных на линиях трансформаторов напряжения, указанных в программе. Это значит, что все его методы и сам ФБ запрограммированы на получение сигналов и их обработку только с выключателей по линии понижающего трансформатора, таких как В 220 Т-2. Подключить к данному ФБ заземляющий нож или разъединитель невозможно.

Также как и вышеперечисленные ФБ, HVCB_T содержит полный набор данных, необходимых для работы. Ниже приведен список локальных переменных функционального блока HVCB_T.

```
FUNCTION_BLOCK HVCB_T IMPLEMENTS Device
VAR_INPUT
  inline:BOOL;
END_VAR
VAR_OUTPUT
  outline:BOOL;
END_VAR
VAR
  myName:          STRING;
  myState:         STRING:='off';
  myMessage:      STRING;
  myDC_T:         POINTER TO DC_T;
  myActiv:        POINTER TO BOOL;
  myDet_vkl:      POINTER TO BOOL;
  myDet_otkl:     POINTER TO BOOL;
  timer:TON;
END_VAR
```


По списку локальных переменных видно, что условия для включения и отключения у данного ФБ отличаются от других.

ФБ имеет другие привязанные КА, ведь для его условий включения и отключения, необходимы КА которые влияют на выключатель, это только разъединитель. Заземляющие ножи тоже влияют на работу выключателя, но так как они зависят от положения своего разъединителя, то в данном случае достаточно знать положение разъединителя, установленного на линии понижающего трансформатора.

Методы и их программный код данного ФБ идентичны методам других ФБ, за исключением метода «TurnOff». Он не содержит условий отключения, так как выключатель должен отключаться в первую очередь при аварийных событиях и выводе линии из работы.

Ниже указан программный код метода «TurnOff». Полный код ФБ и его методов указан в приложении Ж.

```
IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    TurnOff:=TRUE;
END_IF
```

Данный код привязан к состоянию выключателя. Если выключателей находится в состоянии on, то условий для отключения, как в методах других ФБ, здесь нет. При вызове метода состояние автоматически меняется на «отключение».

Функциональный блок «ГКН»: - данный функциональный блок отвечает за работу секционных заземляющих ножей. Это значит, что все его методы и сам ФБ запрограммированы на получение сигналов и их обработку только с ЗНШ РТН-2-220 и ЗНШ ШР-2-220 СВ. Подключить к данному ФБ выключатели или разъединители невозможно.

Функциональные блоки содержат полный набор данных, необходимых для их работы. Ниже приведен список локальных переменных функционального блока «GKH».

```
FUNCTION_BLOCK GKH IMPLEMENTS Device
VAR_INPUT
  inline:BOOL;
END_VAR
VAR_OUTPUT
  outline:BOOL;
END_VAR
VAR
  myName:          STRING;
  myState:         STRING:='off';
  myMessage:      STRING:='';
  myDC:           Device;
  myActiv:        POINTER TO BOOL;
  myDet_vkl:      POINTER TO BOOL;
  myDet_otkl:     POINTER TO BOOL;
  myTransformator: POINTER TO BOOL;
  myDefect_circuit: POINTER TO BOOL;
  command:        STRING; //????????????
  timer:          TON;
END_VAR
```

По списку локальных переменных видно, что условия для включения и отключения у данного ФБ отличаются от других.

ФБ имеет другие привязанные КА, ведь для его условий включения и отключения, необходимы КА которые влияют на заземляющие ножи, это только разъединитель, а также сигнал с измерительного трансформатора тока (ТТ).

Методы и их программный код данного ФБ идентичны методам других ФБ, за исключением условий включения и отключения. Полный код ФБ и его методов указан в приложении Ж.

Остальные функциональные блоки КА, такие как: DC_T (разъединители на линиях понижающих трансформаторов), GK (Заземляющие ножи разъединителей), HVCB_SV (секционный выключатель), HVCB_UKT (выключатель на линии Юктали), RTN (разъединитель измерительного трансформатора тока) рассматривать нет необходимости ведь их содержание схоже со всеми вышеперечисленными функциональными блоками, отличаются лишь условия включения и

отключения. Полный код всех ФБ коммутационных аппаратов и их методов указан в приложение Ж.

Функциональный блок «PlanExecutor» - данный блок является планом исполнения всей программы.

ФБ производит поиск выбранного в подсистеме «формирования и хранения режима переключений» необходимого КА. Так как каждый ФБ имеет свои методы, то при нахождении выбранного КА, программа вызывает его ФБ, а затем его метод «TurnOn» или «TurnOff» в зависимости от того какая операция была выбрана. Ниже указан программный код данного ФБ.

```
FUNCTION_BLOCK PlanExecutor
VAR_INPUT
    IN:BOOL;
//Массив КА
    peMassiv_KA: ARRAY[1..31] OF DEVICE;
END_VAR
-----
IF n>Card.nTask THEN
    IN:=FALSE;
    n:=1;
    i:=1;
    Done:=TRUE;
ELSE Done:=FALSE;
END_IF
IF IN THEN
    WHILE peMassiv_KA[i].Name <> Card.Tasks[n].Object DO
        i:=i+1;
    END_WHILE
    IF peMassiv_KA[i].Name = Card.Tasks[n].Object THEN
        IF Card.Tasks[n].WorkType = 'Включить'
            AND peMassiv_KA[i].State = 'off' THEN
            peMassiv_KA[i].TurnOn();
        ELSIF Card.Tasks[n].WorkType = 'Отключить'
            AND peMassiv_KA[i].State = 'on' THEN
            peMassiv_KA[i].TurnOff();
        ELSIF (Card.Tasks[n].WorkType = 'Отключить'
            AND peMassiv_KA[i].State = 'off') OR
            (Card.Tasks[n].WorkType = 'Включить'
            AND peMassiv_KA[i].State = 'on') THEN
            n:=n+1;
            i:=1;
        END_IF
    END_IF
END_IF
```

```
END_IF  
END_IF
```

Работа ФБ осуществляется следующим образом:

Весь программный код данного ФБ, построен на сравнении идентификаторов аппаратов. Каждый КА имеет свой ФБ, и конечно же свой идентификатор. Поэтому данный код сравнивает идентификаторы КА, содержащихся в массиве устройств `reMassiv_KA`, с компонентом `Object`, структуры `WorkCard` (подробнее о структуре рассказывается в пункте 4.2.1).

Если переменная `IN` принимает значения истины, то в работу запускается два цикла.

Первый цикл «WHILE», он увеличивает индекс массива устройств (в данном случае это переменная `i`), пока идентификаторы КА не идентичны друг другу.

Второй цикл «IF». После нахождения необходимого КА, т.е. при совпадении идентификаторов `reMassiv_KA` и `WorkCard`, определяется операция для найденного КА, она также содержится в структуре `WorkCard`. Если операция «включить» и состояние КА «отключен», то вызывается метод `TurnOn` функционального блока найденного КА. Если же операция «отключить» и состояние КА «включен», то вызывается метод `TurnOff` функционального блока найденного КА. После выполнения операции включения или отключения увеличивается индекс массива структур `Tasks` (в данном случае это переменная `n`) и происходит поиск следующего коммутационного аппарата.

Циклы завершаются в том случае когда будут произведены все операции над КА, выбранными в подсистеме формирования и хранения режима переключений.

4.2.3 Компоненты организации программы (POU)

Компоненты организации программы POU в CODESYS 3.5 являются программами, программного продукта, поэтому далее по тексту, они упоминаются как «программа».

POU «Initialization» - данная программа необходима для инициализации всех КА. Т.е. к каждой переменной, которая имеет тип Device, привязывается свой тип ФБ. Данная программа вызывается однажды при старте системы. Ниже показан пример инициализации КА – ЛР 220 Юктали. Полный программный код указан в Приложении Ж.

```
GVL.LR_220_UKTALI.Init
(
Name:='ЛР 220 Юктали',pHVCB_UKT:=ADR(GVL.V_220_UKTALI),
pGKL:=ADR(GVL.ZNL_LR_220_UKTALI),pGKV:=ADR(GVL.ZNV_LR_220_UKTALI),
pActiv:=ADR(GVL.LR_220_UKTALI_act),
pDet_vkl:=ADR(GVL.Isp_Mech[1].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[1].OFF),
pTransformator:=ADR(GVL.LTransf),
pDefect_circuit:=ADR(GVL.LR_220_UKTALI_CIR),
pDefect_VBU:=ADR(GVL.LR_220_UKTALI_VBU)
);
```

Объявление КА в глобальных переменных:

```
LR_220_UKTALI:          DC_LR;
```

Данный код программы, вызывает метод Init функционального блока DC_LR. В данном случае DC_LR это тип переменной LR_220_UKTALI. В переменные метода записываются глобальные переменные, которые на подстанции являются входами терминала управления. На данные входы приходят все те сигналы, которые упоминались при разборе функционального блока DC_LR.

POU «PLC_PRG» - данная программа является «основной». В ней происходит: запуск всей программы в работу, вызывается POU «Initialization», объявляется массив устройств (Massiv_KA), а также содержатся второстепенные процессы программы. Под второстепенными процессами понимается: перезапись массива Objects, в зависимости от выбранной ячейки подстанции, данный массив отображается в выпадающем списке «Объект» в интерфейсе формирования режима переключений; рассылка массива устройств (Massiv_KA) в другие программы, такие как IM и PlanExecutor; вызов диалогового окна «Ошибка» и «Завершение программы». Программный код данной программы указан в приложении Ж.

POU «IM» - данная программа имитирует работу привода коммутационных аппаратов. Программный код программы указан ниже.

```
PROGRAM IM
VAR_INPUT
    imMassiv_KA: ARRAY[1..31] OF DEVICE;
    full_time: REAL:=10;
END_VAR
VAR_OUTPUT
END_VAR
VAR
    speed: REAL;
END_VAR
-----
IF Isp_Mech[i].position<0.5 THEN
    Isp_Mech[i].OFF:=TRUE;
    Isp_Mech[i].ON:=FALSE;
    IF imMassiv_KA[i].State='GoOn'
        AND Isp_Mech[i].Work_Mode THEN
        speed:=100/full_time;
    ELSE speed:=0;
    END_IF
ELSIF Isp_Mech[i].position>99.5 THEN
    Isp_Mech[i].ON:=TRUE;
    Isp_Mech[i].OFF:=FALSE;
    IF imMassiv_KA[i].State='GoOff'
        AND NOT Isp_Mech[i].Work_Mode THEN
        speed:=-100/full_time;
    ELSE speed:=0;
    END_IF
ELSE Isp_Mech[i].OFF:=FALSE; Isp_Mech[i].ON:=FALSE;
    IF imMassiv_KA[i].State='GoOff' AND NOT Isp_Mech[i].Work_Mode
        THEN speed:=-100/full_time;
    ELSIF imMassiv_KA[i].State='GoOn' AND Isp_Mech[i].Work_Mode
        THEN speed:=100/full_time;
    ELSE speed:=0;
    END_IF
END_IF
Isp_Mech[i].integ(IN:=speed, TM:=20,RESET:=FALSE,
    OUT=>Isp_Mech[i].position);
```

В коде используется структура Isp_Mech (о данной структуре подробно рассказывается в пункте 4.2.1).

Работа программы организована следующим образом:

Если КА находится в отключенном положении, то переменная position будет меньше 0,5 и компонент OFF структуры Isp_Mech будет иметь истинное значение. Если КА находится во включенном положении, то переменная position будет больше 99,5 и компонент ON структуры Isp_Mech будет иметь истинное значение. Иначе обе переменные будут иметь значение – ложь.

Сигнал о включение приходит от массива устройств imMassiv_KA. У данного массива вызывается свойство State, и если состояние КА «включение» или «отключение», то привод включается в работу.

Прототип программы был реализован в CODESYS 3.5 и опробован на виртуальном контроллере CODESYS Control Win V3.

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

5.1 Безопасность

В данном разделе описывается человеко-машинный интерфейс автоматизированной программы переключений, то насколько он безопасен и целесообразно ли его использовать на производстве.

Интерфейс состоит из двух отдельных окон:

Первое окно – меню формирования режима переключений.

Второе окно – меню управления.

5.1.1 Меню формирования режима переключений

В данном меню, происходит выбор коммутирующих аппаратов, выбор операции над коммутирующим аппаратом (включение/отключение), а также задаётся последовательность выполнения команд.

Но также данный интерфейс может сохранять созданные пользователем шаблоны, и загружать ранее сохранённые шаблоны.

Данное меню представлено на рисунке 23.

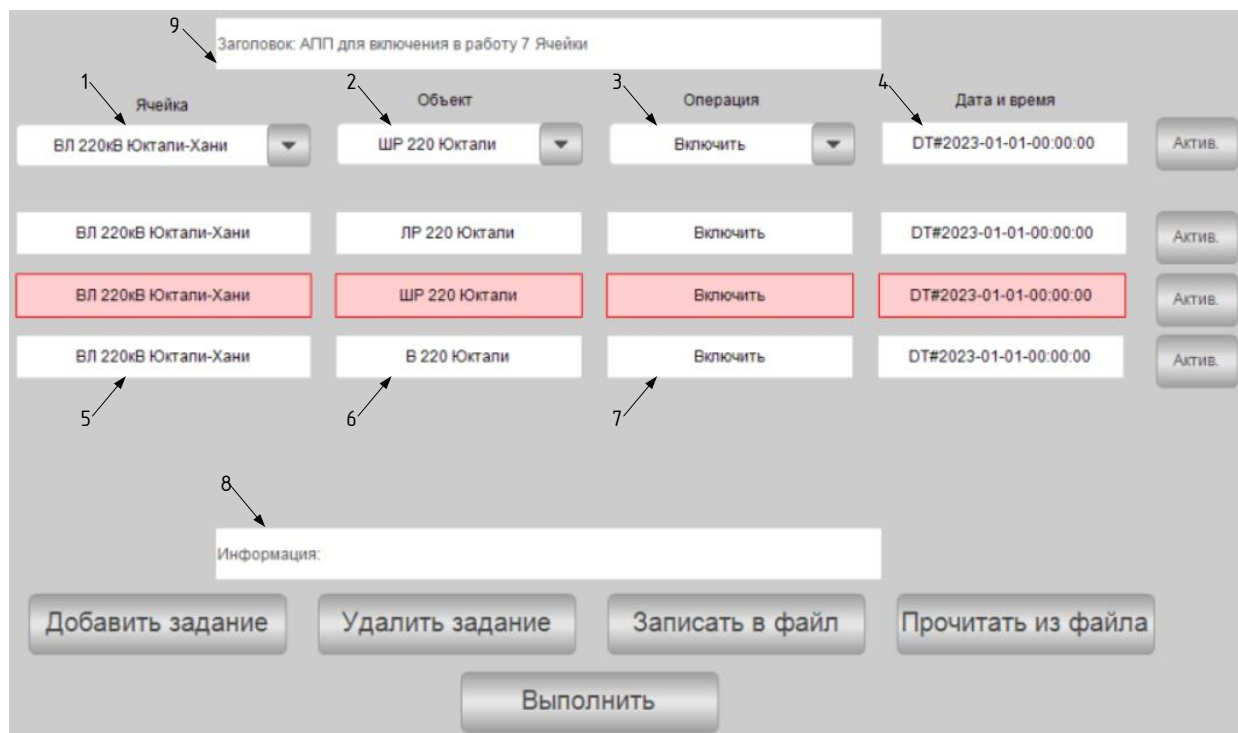


Рисунок 23 – Меню формирования режима переключений

Меню состоит из 3-ёх выпадающих списков, текстового поля – дата и время, кнопок управления, полей, в которые заносятся данные, выбранные в выпадающих списках, а также два дополнительных текстовых поля – «Заголовок» и «Информация».

Под номером 1 указан выпадающий список содержащий список ячеек подстанции. Данный список имеет свойство изменять выпадающий список под номером 2. Весь список показан на рисунке 24.

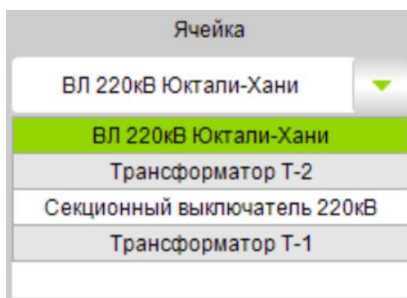


Рисунок 24 – Выпадающий список ячеек подстанции

Под номером 2 указан выпадающий список коммутационных аппаратов, установленных на выбранной ячейке подстанции. Весь список показан на рисунке 25.



Рисунок 25 – Выпадающий список коммутационных аппаратов

Под номером 3 указан выпадающий список операций, проводимые над коммутирующим аппаратом. Весь список показан на рисунке 26.

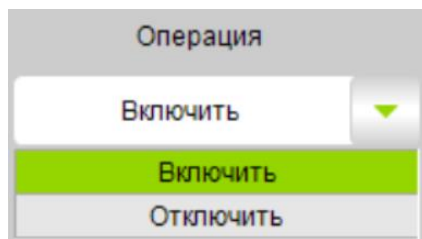


Рисунок 26 – Выпадающий список операций

Под номером 4 указано текстовое поле «Дата и время». В данное поле записывается точное время и дата переключения коммутирующего аппарата. Количество текстовых полей может быть не более пяти.

Под номером 5 указано текстовое поле, в которое заносится ячейка подстанции, выбранная в выпадающем списке 1. Количество текстовых полей может быть не более пяти. Это значит, что программа может работать максимум с 5 коммутирующими аппаратами и минимум с 1 коммутирующим аппаратом.

Под номером 6 указано текстовое поле, в которое заносится название коммутирующего аппарата, выбранного в выпадающем списке 2. Количество текстовых полей может быть не более пяти.

Под номером 7 указано текстовое поле, в которое заносится операция над коммутирующим аппаратом, выбранная в выпадающем списке 3. Количество текстовых полей может быть не более пяти.

Под номером 8 указано текстовое поле «Информация». В данное поле вносится вся дополнительная информация, например Ф.И.О. того кто составил шаблон переключений, или же при ошибке во время переключений, записал в данное поле в чём заключалась ошибка переключений.

Под номером 9 указано текстовое поле «Заголовок». В него заносится названия всей программы переключений. Т.к. в ячейке всего 3 основных коммутационных аппарата, то программа сможет как отключить их последовательно, так и включить, а это значит, что при помощи программы можно отключить целую ячейку подстанции. Для этого и необходим заголовок, в котором будет написано к примеру - «АПП для включения/отключения 7 Ячейки».

Кнопка «Добавить задание». Данная кнопка добавляет строку, состоящую из текстовых полей: 4,5,6,7.

Кнопка «Удалить задание». Данная кнопка удаляет строку, созданную кнопкой «Добавить задание».

Кнопка «Записать в файл». Данная кнопка создаёт файл, в котором будет храниться готовый шаблон автоматизированной программы переключений. Т.е. в

данном файле будут сохранены все выбранные ячейки, необходимые коммутирующие аппараты и операции над ними, а также заголовок и информацию.

Кнопка «Прочитать из файла». Данная кнопка позволяет открыть файл, содержащий в себе шаблон АПП, созданный кнопкой «Записать в файл».

Кнопка «Актив.» (Активировать). Данная кнопка заносит в текстовые поля: 4, 5, 6, 7 данные, выбранные в выпадающих списках 1, 2, 3. Каждая кнопка «Актив.» заносит информацию в ту строку, напротив которой она находится. Но кнопка «Актив.» находящаяся напротив выпадающих списков, заносит выбранную информацию в те поля, которые были выбраны при помощи мыши.

Кнопка «Выполнить». Данная кнопка переносит пользователя в меню управления.

5.1.2 Меню управления

В данном меню происходит запуск программы, и отображение её выполнения. Данное меню представлено на рисунке 19 в подпункте 4.1.1.

Само меню или интерфейс, состоит из:

1) Схемы. Данная схема отображает переключение выбранных коммутирующих аппаратов.

2) Консоли управления программой. Данная консоль необходима для задания подтверждающих сигналов, необходимых для удовлетворения условий переключения. Такими сигналами являются: «Сигналы с трансформаторов» и «Неисправности». «Дистанционный режим управления» не является таким сигналом.

Подробнее о консоли управления программой:

Раздел «Дистанционный режим управления». В данном разделе происходит переключение режима управления коммутирующими аппаратами. Т.к. одной ячейкой управляет один шкаф УСО, следовательно в одном таком шкафу стоит внутренний блок управления, который переводит работу ячейки с местного режима управления на дистанционный и наоборот. Кнопки в данном разделе и управляют устройством ВБУ.

Раздел «Сигналы с трансформаторов». Это подтверждающие сигналы, имитирующие сигналы с измерительных трансформаторов тока и напряжения. Необходимы для формирования условий переключения.

Раздел «Неисправности». Это подтверждающие сигналы, имитирующие неисправности у коммутационных аппаратов. Необходимы для проверки правильности работы программы.

Кнопка «Запустить программу». Данная кнопка запускает программу на исполнение.

Кнопка «Вернуться в меню». Данная кнопка переносит пользователя в меню формирования режима переключений.

Кнопка «Проверка ошибки». Данная кнопка выводит окно, в котором будет указана ошибка переключения. Также данное окно откроется автоматически в случае ошибки переключений. Окно представлено на рисунке 27.



Рисунок 27 – Окно ошибок

В итоге, весь интерфейс исключает возможность ошибки самого оператора, и поэтому является безопасным для работы с коммутирующими аппаратами.

5.2 Экологичность

В данном разделе будет рассматриваться автоматизированное рабочее место (далее - АРМ) инженера АСУ. Что входит в его состав, какие поломки могут произойти и как утилизируют устройства неподлежащие ремонту.

Состав АРМ:

- 1) Мебель (Столы и стулья).
- 2) Монитор 23,8” 1920x1200 LED AC220V.
- 3) Принтер отчетов (лазерный, ч/б, А3/А4).
- 4) Манипулятор «мышь» оптическая USB.
- 5) Клавиатура USB.
- 6) Сетевой фильтр (8 розеток).
- 7) Стерео колонки для звуковой сигнализации.

Неисправности основных частей АРМа:

К неисправностям монитора относятся: мигание изображения, монитор не включается, внезапное отключение, вертикальные разноцветные полосы, чёрные пятна, сниженная яркость и дрожание изображения.

К неисправностям лазерного принтера относятся: мнётся бумага, принтер не печатает, при печати оставляет вертикальные чёрные и белые полосы, мнётся бумага, принтер не включается, оставляет тёмные и бледные отпечатки, при работе издаёт скрежет и скрип, внезапное отключение, остановка бумаги.

К неисправностям «мыши» относятся: мышь не реагирует на подключение, курсор мыши прыгает или движется рывками, колёсико прокрутки не работает или при прокрутке область двигается рывками, левая или правая клавиша западают или не реагируют на нажатие, а также делают двойное нажатие.

К неисправностям клавиатуры относятся: повреждение кнопок, клавиатура не реагирует на подключение, неправильный вывод информации на монитор, западание кнопок клавиатуры.

К неисправностям стерео колонок относятся: выход из строя динамиков, поломка блока питания, поломка регулятора громкости.

Если одна из частей АРМа не подлежит ремонту, то она утилизируется. Утилизация проходит в несколько этапов:

- 1) Заказчик заключает договор с исполнителем.
- 2) Состоится вывоз оргтехники с предприятия.
- 3) Исполнитель демонтирует, сортирует технику. Отделяет черный металл от цветного и драгметаллов.

4) Полученное сырье отправляется на заводы для переработки. В дальнейшем из них будут сделаны новые продукты.

5) Отходы классов повышенной опасности обезвреживаются и уничтожаются, либо их отвозят на легальные места захоронения.

6) Заказчик получает акт выполненных работ вместе с необходимыми для бухучета документами.

5.3 Чрезвычайные ситуации

К чрезвычайным ситуациям относится пожар на АРМе. В данном разделе будут рассматриваться причины пожара и действия персонала при пожаре.

Причинами пожара могут стать:

1) Короткие замыкания. Они возникают из-за перенапряжений в сети, поврежденной изоляции, старой проводкой или нарушениями правил ее эксплуатации.

2) Использование неисправного электрооборудования. Выключатели, розетки, техника с поврежденной изоляцией или неисправные приборы.

3) Применение обогревательных приборов открытого типа.

В случае пожара, действия работников организации и привлекаемых к тушению лиц в первую очередь должны быть направлены на обеспечение безопасности людей, их эвакуацию и спасение.

Каждый работник организации, обнаруживший пожар и его признаки (задымление, запах горения или тления различных материалов, повышение температуры и т.п.) обязан:

1. Немедленно сообщить о пожаре в пожарную охрану по городскому телефону по номеру «01» или по мобильному телефону по номеру «112» (при этом необходимо четко назвать адрес организации, место возникновения пожара, а также сообщить свою должность и фамилию);

2. Известить о пожаре руководителя организации или замещающего его работника;

3. Задействовать систему оповещения людей о пожаре, приступить самому и привлечь других лиц к эвакуации из здания в безопасное место согласно плану эвакуации;

4. При необходимости отключить электроэнергию, приостановить работу отдельных агрегатов и участков, способствующих развитию пожара и задымлению помещений здания;

5. Оценить обстановку и приступить к тушению очага возгорания имеющимися первичными средствами пожаротушения (огнетушителями), для ликвидации его на ранней стадии;

6. Организовать встречу пожарных подразделений (выделить для встречи пожарных подразделений лицо, хорошо знающее расположение подъездных путей и водоисточников).

ЗАКЛЮЧЕНИЕ

При выполнении работы были закреплены и углублены знания, полученные в процессе обучения.

В результате выполнения работы были решены следующие задачи:

- изучен объект автоматизации;
- рассмотрена структура существующей системы управления;
- произведено описание объектов автоматизации;
- изучены схемы управления коммутационными аппаратами;
- разработана автоматизированная программа переключений;
- спроектирован интерфейс управления программой.

Проведенная проверка автоматизированной программы переключений, позволила выявить, что программа работает правильно и соответствует требованиям, указанным в задании.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. СТО 56947007- 25.040.40.227-2016. Типовые технические требования к функциональной структуре автоматизированных систем управления технологическими процессами подстанций Единой национальной электрической сети.

2. Автоматизированные системы управления технологическими процессами на базе ПТК EVICON: Издание 2, ЕКРА_РТК-EVICON_2016. [Электронный ресурс]. – Режим доступа: https://ekra.ru/upload/iblock/36e/ЕКРА_РТК-EVICON_2016.pdf (последняя дата обращения 20.06.2023).

3. Инструкция по эксплуатации и оперативному обслуживанию разъединителей РГН-220кВ, РГН-110 кВ. ПС 220 кВ. [Электронный ресурс]. – Режим доступа: <https://stroystandart.info/index.php?name=files&op=view&id=5239> (последняя дата обращения 20.06.2023).

4. АВВ. Выключатель HPL-245 В1. [Электронный ресурс]. – Режим доступа: <http://eknis.net/uploads/files/sf6-switch.pdf> (последняя дата обращения 30.03.2023).

5. УЭТМ. Выключатели элегазовые серии ВГТ-УЭТМ на 35, 110 и 220 кВ. [Электронный ресурс]. – Режим доступа: <https://dv-electro.ru/images/opros/VGT-35,110,220.pdf> (последняя дата обращения 20.06.2023).

6. Руководство пользователя. CODESYS V3.5. Первый старт версия 3.0. 2022. [Электронный ресурс]. – Режим доступа: https://ftp.owen.ru/CoDeSys3/11_Documentation/03_3.5.11.5/CDSv3.5_FirstStart_v3.0.pdf (последняя дата обращения 20.06.2023).

7. Руководство по эксплуатации ЭКРА 24Х. [Электронный ресурс]. – Режим доступа: <https://ekra.ru/product/docs/upravlenie-prisoed/kontrollery-prisoed/Руководство%20по%20эксплуатации%20ЭКРА%2024Х.pdf> (последняя дата обращения 20.06.2023).

8. Руководство пользователя для контроллера Advantech ECU-4784 Series. [Электронный ресурс]. – Режим доступа: https://nnz-ipc.ru/files/documentation/advantech/um-ecu-4784_manual.pdf (последняя дата обращения 20.06.2023).

9. Проектирование систем противопожарной защиты на объектах ОАО «ФСК ЕЭС». Общие технические требования. [Электронный ресурс]. – Режим доступа: https://www.fsk-ees.ru/upload/docs/03.82_sto_56947007-33.040.10.139-2012_red.pdf (последняя дата обращения 20.06.2023).

10. СТО 25.040.40.226-2016. Общие технические требования к АСУТП ПС ЕНЭС. Основные требования к программно-техническим средствам и комплексам (последняя дата обращения 20.06.2023).

11. Инструкция по охране труда для диспетчера. [Электронный ресурс]. – Режим доступа: https://ohranatruda.ru/ot_biblio/instructions/165/3251/ (последняя дата обращения 20.06.2023).

12. Булгаков А.Б. Безопасность жизнедеятельности [Электронный ресурс]. – Режим доступа: https://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/6979.pdf (последняя дата обращения 20.06.2023).

13. Рыбалёв, А. Н. Разработка и исследование алгоритмов и программ управления исполнительными механизмами систем автоматического регулирования. [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/item.asp?id=18969993> (последняя дата обращения 20.06.2023).

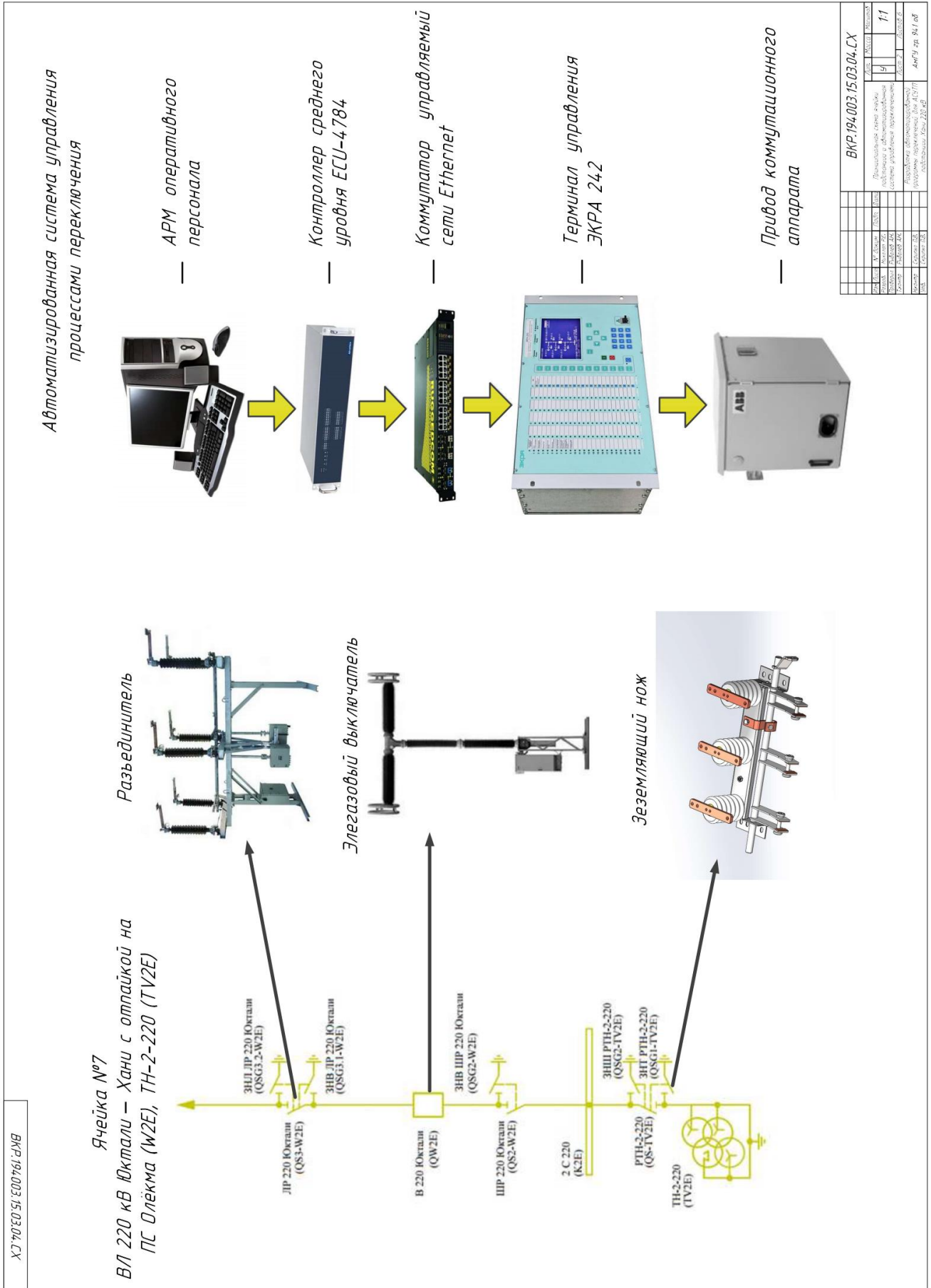
14. Привод электродвигательный ПД-14М. [Электронный ресурс]. – Режим доступа: <http://www.razrad.ru/wp-content/themes/storefront-child/docs/privody/pd14.pdf> (последняя дата обращения 20.06.2023).

15. Codesys ООП. Объектно-ориентированное программирование на ПЛК в АСУТП. Часть 1. Промышленное программирование и автоматизация [Электронный ресурс]. – Режим доступа: <https://blog.engcore.ru/2021/12/11/oop-plc-part1/> (последняя дата обращения 20.06.2023).

16. Codesys. Объектно-ориентированное программирование на ПЛК. Часть 2. Классы и интерфейсы, методы и свойства. Промышленное программирование и автоматизация [Электронный ресурс]. – Режим доступа: <https://blog.engcore.ru/2021/12/20/codesys-oop-plc-part2/>. (последняя дата обращения 20.06.2023).

ПРИЛОЖЕНИЕ Б

Принципиальная схема ячейки подстанции и АСУ переключениями



Организационная структура АПП

Структура WorkCard

Title АПП для включения в работу 7 Ячейки

Tasks 1 2 3 4 5 6

nTask 3

Information Составитель: Иванов М.Е.

Структура Task

Cell Ячейка 7

Object ЛР 220 Юкталли

WorkType Включить

DataAndTime 2023-01-01-00:00:00

Список локальных переменных функционального блока DC_T

```

FUNCTION BLOCK DC_T IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:
        STRING;
    myState:
        STRING:= 'off';
    myHVCB_T, myGKT, myGKV:
        DEVICE
    myActiv, myDet_vkl, myDet_otkl:
        POINTER TO BOOL;
    myDefect_circuit, myDefect_VBU:
        POINTER TO BOOL;
    timer:TON;
END_VAR
                
```

Объектно-ориентированное программирование

Интерфейс «Устройство» -> O Device

Разъединители

Выключатель трансформаторов

Выключатель линии Юкталли

Секционный выключатель

Заземляющие ножи

Линейный разъединитель

Общая информация:

Объект: АПП для включения в работу 7 Ячейки

Структура: DC_T

Составитель: Иванов М.Е.

Дата: 2023-01-01 00:00:00

ВКР: 194.003.15.03.04.80

Организационная структура АПП

Лист: 6 из 11

Англ. стр. 9/1 об.

ПРИЛОЖЕНИЕ Ж

Программный код

```
VAR_GLOBAL
Card: WorkCard;
cTask: BYTE:=0;
Objects: ARRAY[0..10] OF STRING(30):=['ЗНЛ ЛР 220 Юктали','ЛР 220 Юктали','ЗНВ ЛР 220 Юктали',
'В 220 Юктали','ЗНВ ШР 220 Юктали','ШР 220 Юктали','ЗНШ РТН-2-220','РТН-2-220','ЗНТ РТН-2-220'];
nObject: BYTE:=0;
WorkTypes: ARRAY[0..1] OF STRING(10):=['Включить','Отключить'];
nWorkType: BYTE:=0;
DateAndTime: DT:=DT#2023-01-01-00:00:00;
Cells: ARRAY [0..3] OF STRING(30):= ['ВЛ 220кВ Юктали-Хани','Трансформатор Т-2','Секционный выключатель
220кВ','Трансформатор Т-1'];
nCell: BYTE:=0;
fbOpenDialog_fault: VU.FbOpenDialog;//ФБ для открытия диалогового окна ошибки
fbOpenDialog_Done: VU.FbOpenDialog;//ФБ для открытия диалогового окна выполнения программы
stDialogCoordinates: VisuElems.CmpVisuHandler.VisuStructPoint := (iX := 200, iY := 100);
// _____ Сигналы о включении/отключении устройств _____
LR_220_UKTALI_act, TR_220_T_1_act, TR_220_T_2_act, HR_220_UKTALI_act, HR_220_T_1_act, HR_220_T_2_act, RTN_2_220_act,
RTN_1_220_act, V_220_UKTALI_act, V_220_T_1_act, V_220_T_2_act, SV_220_act, HR_1_220_SV_act, HR_2_220_SV_act,
ZNH_HR_1_220_SV_act, ZNH_HR_2_220_SV_act, ZNV_HR_1_220_SV_act, ZNV_HR_2_220_SV_act, ZNL_LR_220_UKTALI_act,
ZNV_LR_220_UKTALI_act, ZNV_HR_220_UKTALI_act, ZNH_RTN_2_220_act, ZNT_RTN_2_220_act, ZNT_TR_220_T_2_act,
ZNV_TR_220_T_2_act, ZNV_HR_220_T_2_act, ZNH_RTN_1_220_act, ZNT_RTN_1_220_act, ZNT_TR_220_T_1_act, ZNV_TR_220_T_1_act,
ZNV_HR_220_T_1_act: BOOL:=FALSE;
// _____ Неисправности _____
Fault_VBU, Fault_circuit: BOOL;
Dialog_fault: BOOL;// переменная для вывода диалогового окна ошибки
// _____ РЕЖИМЫ УПРАВЛЕНИЯ и СИГНАЛЫ ПОДТВЕРЖДЕНИЯ _____
Transf, LTransf: BOOL;// сигналы с трансформаторов
PUSK: BOOL;//переменная отвечающая за запуск программы
VL_Uktali: BOOL;//переменная переключателя режима управления 7 ячейки
Transformer_1: BOOL;//переменная переключателя режима управления 4 ячейки
Transformer_2: BOOL;//переменная переключателя режима управления 6 ячейки
SV_HVCB: BOOL;//переменная переключателя режима управления 5 ячейки
Done: BOOL;//переменная для вывода диалогового окна выполнения программы
i: BYTE (1..31);//индекс массива коммутационных аппаратов
n: BYTE (0..5);//индекс массива структуры Card
Isp_Mech: ARRAY [1..31] OF I_Mech;// массив КА для программы IM
Fault_Message: ARRAY [1..31] OF Dialog;// массив для диалогового окна ошибки
// _____ Неисправности по ВБУ _____
LR_220_UKTALI_VBU, TR_220_T_1_VBU, TR_220_T_2_VBU, HR_220_UKTALI_VBU, HR_220_T_1_VBU, HR_220_T_2_VBU,
RTN_2_220_VBU, RTN_1_220_VBU, HR_1_220_SV_VBU, HR_2_220_SV_VBU, ZNH_HR_2_220_SV_VBU, ZNV_HR_2_220_SV_VBU,
ZNV_HR_220_T_2_VBU : BOOL:=FALSE;
// _____ Неисправности цепей питания и обогрева _____
LR_220_UKTALI_CIR, TR_220_T_1_CIR, TR_220_T_2_CIR, HR_220_UKTALI_CIR, HR_220_T_1_CIR, HR_220_T_2_CIR, RTN_2_220_CIR,
RTN_1_220_CIR, HR_1_220_SV_CIR, HR_2_220_SV_CIR, ZNH_HR_1_220_SV_CIR, ZNH_HR_2_220_SV_CIR, ZNV_HR_1_220_SV_CIR,
ZNV_HR_2_220_SV_CIR, ZNL_LR_220_UKTALI_CIR, ZNV_LR_220_UKTALI_CIR, ZNV_HR_220_UKTALI_CIR, ZNH_RTN_2_220_CIR,
ZNT_RTN_2_220_CIR, ZNT_TR_220_T_2_CIR, ZNV_TR_220_T_2_CIR, ZNV_HR_220_T_2_CIR, ZNH_RTN_1_220_CIR,
ZNT_RTN_1_220_CIR, ZNT_TR_220_T_1_CIR, ZNV_TR_220_T_1_CIR, ZNV_HR_220_T_1_CIR: BOOL:=FALSE;
// _____ ФУНКЦИОНАЛЬНЫЕ БЛОКИ - 31 _____
LR_220_UKTALI: DC_LR;
TR_220_T_1: DC_T;
TR_220_T_2: DC_T;
V_220_UKTALI: HVCB_UKT;
V_220_T_1: HVCB_T;
V_220_T_2: HVCB_T;
SV_220: HVCB_SV;
HR_220_UKTALI: HR;
HR_220_T_1: HR;
HR_220_T_2: HR;
HR_1_220_SV: HR;
HR_2_220_SV: HR;
RTN_2_220: RTN;
RTN_1_220: RTN;
ZNH_HR_1_220_SV: GKH;
ZNH_HR_2_220_SV: GKH;
```

Продолжение Приложения Ж

```

ZNL_LR_220_UKTALI:      GK;
ZNV_LR_220_UKTALI:      GK;
ZNV_HR_220_UKTALI:      GK;
ZNH_RTN_2_220:          GKH;
ZNT_RTN_2_220:          GK;
ZNT_TR_220_T_2:         GK;
ZNV_TR_220_T_2:         GK;
ZNV_HR_220_T_2:         GK;
ZNV_HR_2_220_SV:        GK;
ZNV_HR_1_220_SV:        GK;
ZNH_RTN_1_220:          GKH;
ZNT_RTN_1_220:          GK;
ZNT_TR_220_T_1:         GK;
ZNV_TR_220_T_1:         GK;
ZNV_HR_220_T_1:         GK;
// Работа с файлами
fbdFLP:                  VisuDialogs.Visu_FbFileListProvider;
stDir, stFile, fullPath: STRING;
//Возвращает код ошибки системы
rte:                     VisuDialogs.SysFile.SysTypes.RTS_IEC_RESULT;
//Указатель на файл
FileD:                   VisuDialogs.SysFile.SysTypes.RTS_IEC_HANDLE;
updating:                BOOL:=FALSE;
END_VAR

TYPE Dialog :
STRUCT
    Name:    STRING;//идентификатор коммутационного аппарата
    Message: STRING;//сообщение от коммутационного аппарата
END_STRUCT
END_TYPE

TYPE I_Mech :
STRUCT
    Work_Mode:    BOOL;//Режим работы привода
    ON:           BOOL:=FALSE;//состояние привода «включен»
    OFF:          BOOL:=TRUE;//состояние привода «отключен»
    position:     REAL:=0;//позиция привода
    integ:        INTEGRAL;//ФБ имитирующий электромотор привода
END_STRUCT
END_TYPE

TYPE Task :
STRUCT
    Cell:         STRING(30);//переменная для записи выбранной ячейки
    Object:       STRING(20);//переменная для записи выбранного КА
    WorkType:     STRING(20);///переменная для записи выбранной операции
    DateAndTime: DT;//переменная для записи даты и времени переключения
END_STRUCT
END_TYPE

TYPE WorkCard :
STRUCT
    Title:        STRING(50);//Заголовок АПП
    Tasks:        ARRAY[1..6] OF Task;//Массив структур содержащий строки АПП
    nTask:        BYTE;//Индекс указывающий на строку АПП
    Information:  STRING(100);//Дополнительная информация для АПП
END_STRUCT
END_TYPE

PROGRAM PLC_PRG
VAR
    isInit:BOOL:=FALSE;//переменная для разовой инициализации ФБ
    PE:      PlanExecutor;//Переменная исполнительного плана
    //массив КА подстанции
    Massiv_KA: ARRAY[1..31] OF DEVICE :=
        [(LR_220_UKTALI),(TR_220_T_1),(TR_220_T_2),(V_220_UKTALI),(V_220_T_1),(V_220_T_2),(SV_220),

```

Продолжение Приложения Ж

```
(HR_220_UKTALI),(HR_220_T_1),(HR_220_T_2),(HR_1_220_SV),(HR_2_220_SV),(RTN_2_220),(RTN_1_220),
(ZNH_HR_1_220_SV),(ZNH_HR_2_220_SV),(ZNL_LR_220_UKTALI),(ZNV_LR_220_UKTALI),(ZNH_RTN_2_220),
(ZNH_RTN_1_220),(ZNT_RTN_2_220),(ZNT_RTN_1_220),(ZNT_TR_220_T_2),(ZNT_TR_220_T_1),
(ZNV_TR_220_T_1),(ZNV_TR_220_T_2),(ZNV_HR_220_T_2),(ZNV_HR_220_T_1),(ZNV_HR_2_220_SV),
(ZNV_HR_1_220_SV),(ZNV_HR_220_UKTALI)];

END_VAR

CASE nCell OF
0:
    Objects[0]:='ЗНЛ ЛР 220 Юктали'; Objects[1]:='ЛР 220 Юктали'; Objects[2]:='ЗНВ ЛР 220 Юктали';
    Objects[3]:='В 220 Юктали'; Objects[4]:='ЗНВ ШП 220 Юктали'; Objects[5]:='ШП 220 Юктали';
    Objects[6]:='ЗНШ ПТН-2-220'; Objects[7]:='РТН-2-220'; Objects[8]:='ЗНТ РТН-2-220'; Objects[9]:='';
1:
    Objects[0]:='ЗНТ ТР 220 Т-2'; Objects[1]:='ТР 220 Т-2'; Objects[2]:='ЗНВ ТР 220 Т-2'; Objects[3]:='В 220 Т-2';
    Objects[4]:='ЗНВ ШП 220 Т-2'; Objects[5]:='ШП 220 Т-2'; Objects[6]:=''; Objects[7]:=''; Objects[8]:='';
    Objects[9]:='';
2:
    Objects[0]:='ЗНШ ШП-1- 220 СВ '; Objects[1]:='ШП-1-220 СВ'; Objects[2]:='ЗНВ ШП-1-220 СВ'; Objects[3]:='СВ 220';
    Objects[4]:='ЗНВ ШП-2-220 СВ'; Objects[5]:='ШП-2-220 СВ'; Objects[6]:='ЗНШ ШП-2-220 СВ';
    Objects[7]:='ЗНШ ПТН-1-220 '; Objects[8]:='РТН-1-220 '; Objects[9]:='ЗНТ РТН-1-220';
3:
    Objects[0]:='ЗНТ ТР 220 Т-1 '; Objects[1]:='ТР 220 Т-1'; Objects[2]:='ЗНВ ТР 220 Т-1'; Objects[3]:='В 220 Т-1';
    Objects[4]:='ЗНВ ШП 220 Т-1'; Objects[5]:='ШП 220 Т-1'; Objects[6]:=''; Objects[7]:=''; Objects[8]:='';
    Objects[9]:='';
END_CASE;

IF NOT isInit THEN
    Initialization();
    isInit:=TRUE;
END_IF

Line(); //вызов программы для постоянной работы ФБ

IM(imMassiv_KA:= Massiv_KA);
PE(peMassiv_KA:= Massiv_KA);
Fault_Message[i].Name:=Massiv_KA[i].Name;
Fault_Message[i].Message:=Massiv_KA[i].Message;

IF PUSK THEN
    PE(IN:=TRUE);
END_IF;

fbOpenDialog_fault
(
    xExecute := (GVL.Dialog_fault=TRUE),
    itfClientFilter := VU.Globals.AllClients,
    sDialogName := 'Dialog_fault',
    xModal := FALSE,
    pTopLeftPosition:=ADR(stDialogCoordinates)
);

fbOpenDialog_Done
(
    xExecute := (GVL.Done=TRUE),
    itfClientFilter := VU.Globals.AllClients,
    sDialogName := 'Done',
    xModal := TRUE,
    pTopLeftPosition:=ADR(stDialogCoordinates)
);

PROGRAM IM
VAR_INPUT
    imMassiv_KA:    ARRAY[1..31] OF DEVICE; //Массив КА
    full_time:     REAL:=10;//время выполнения операций
END_VAR
VAR_OUTPUT
```

Продолжение Приложения Ж

```
END_VAR
VAR
    speed: REAL;//скорость работы привода
END_VAR

IF Isp_Mech[i].position<0.5 THEN
    Isp_Mech[i].OFF:=TRUE;
    Isp_Mech[i].ON:=FALSE;
    IF imMassiv_KA[i].State='GoOn' AND Isp_Mech[i].Work_Mode THEN speed:=100/full_time;
    ELSE speed:=0;
    END_IF
ELSIF Isp_Mech[i].position>99.5 THEN
    Isp_Mech[i].ON:=TRUE;
    Isp_Mech[i].OFF:=FALSE;
    IF imMassiv_KA[i].State='GoOff' AND NOT Isp_Mech[i].Work_Mode THEN speed:=-100/full_time;
    ELSE speed:=0;
    END_IF
ELSE Isp_Mech[i].OFF:=FALSE; Isp_Mech[i].ON:=FALSE;
    IF imMassiv_KA[i].State='GoOff' AND NOT Isp_Mech[i].Work_Mode THEN speed:=-100/full_time;
    ELSIF imMassiv_KA[i].State='GoOn' AND Isp_Mech[i].Work_Mode THEN speed:=100/full_time;
    ELSE speed:=0;
    END_IF
END_IF
Isp_Mech[i].integ(IN:=speed, TM:=20,RESET:=FALSE, OUT=>Isp_Mech[i].position);

PROGRAM Initialization
VAR
END_VAR

GVL.LR_220_UKTALI.Init(Name:='JIP 220 Юктали',pHVCB_UKT:=ADR(GVL.V_220_UKTALI),
    pGKL:=ADR(GVL.ZNL_LR_220_UKTALI),pGKV:=ADR(GVL.ZNV_LR_220_UKTALI),
    pActiv:=ADR(GVL.LR_220_UKTALI_act), pDet_vkl:=ADR(GVL.Isp_Mech[1].ON),
    pDet_otkl:=ADR(GVL.Isp_Mech[1].OFF), pTransformator:=ADR(GVL.LTransf),
    pDefect_circuit:=ADR(GVL.LR_220_UKTALI_CIR),
    pDefect_VBU:=ADR(GVL.LR_220_UKTALI_VBU));
GVL.TR_220_T_1.Init(Name:='TP 220 T-1',pHVCB_T:=ADR(GVL.V_220_T_1),pGKT:=ADR(GVL.ZNT_TR_220_T_1),
    pGKV:=ADR(GVL.ZNV_TR_220_T_1), pActiv:=ADR(GVL.TR_220_T_1_act),
    pDet_vkl:=ADR(GVL.Isp_Mech[2].ON), pDet_otkl:=ADR(GVL.Isp_Mech[2].OFF),
    pDefect_circuit:=ADR(GVL.TR_220_T_1_CIR), pDefect_VBU:=ADR(GVL.TR_220_T_1_VBU));
GVL.TR_220_T_2.Init(Name:='TP 220 T-2',pHVCB_T:=ADR(GVL.V_220_T_2),pGKT:=ADR(GVL.ZNT_TR_220_T_2),
    pGKV:=ADR(GVL.ZNV_TR_220_T_2), pActiv:=ADR(GVL.TR_220_T_2_act),
    pDet_vkl:=ADR(GVL.Isp_Mech[3].ON), pDet_otkl:=ADR(GVL.Isp_Mech[3].OFF),
    pDefect_circuit:=ADR(GVL.TR_220_T_2_CIR), pDefect_VBU:=ADR(GVL.TR_220_T_2_VBU));
GVL.V_220_UKTALI.Init(Name:='B 220 Юктали',pHR_UKT:=ADR(GVL.HR_220_UKTALI),pDC_LR:=ADR(GVL.LR_220_UKTALI),
    pActiv:=ADR(GVL.V_220_UKTALI_act),pDet_vkl:=ADR(GVL.Isp_Mech[4].ON),
    pDet_otkl:=ADR(GVL.Isp_Mech[4].OFF));
GVL.V_220_T_1.Init(Name:='B 220 T-1',pDC_T:=ADR(GVL.TR_220_T_1),pActiv:=ADR(GVL.V_220_T_1_act),
    pDet_vkl:=ADR(GVL.Isp_Mech[5].ON),pDet_otkl:=ADR(GVL.Isp_Mech[5].OFF));
GVL.V_220_T_2.Init(Name:='B 220 T-2',pDC_T:=ADR(GVL.TR_220_T_2),pActiv:=ADR(GVL.V_220_T_2_act),
    pDet_vkl:=ADR(GVL.Isp_Mech[6].ON),pDet_otkl:=ADR(GVL.Isp_Mech[6].OFF));
GVL.SV_220.Init(Name:='CB 220',pHR_SV_1:=ADR(GVL.HR_1_220_SV),pHR_SV_2:=ADR(GVL.HR_2_220_SV),
    pActiv:=ADR(GVL.SV_220_act),pDet_vkl:=ADR(GVL.Isp_Mech[7].ON),
    pDet_otkl:=ADR(GVL.Isp_Mech[7].OFF));
GVL.HR_220_UKTALI.Init(Name:='ШИП 220 Юктали',pHVCB:=GVL.V_220_UKTALI,pGK:=ADR(GVL.ZNV_HR_220_UKTALI),
    pGKH_1:=ADR(GVL.ZNH_HR_2_220_SV), pGKH_2:=ADR(GVL.ZNH_RTN_2_220),
    pActiv:=ADR(GVL.HR_220_UKTALI_act),pDet_vkl:=ADR(GVL.Isp_Mech[8].ON),
    pDet_otkl:=ADR(GVL.Isp_Mech[8].OFF), pDefect_circuit:=ADR(GVL.HR_220_UKTALI_CIR),
    pDefect_VBU:=ADR(GVL.HR_220_UKTALI_VBU));
GVL.HR_220_T_1.Init(Name:='ШИП 220 T-1',pHVCB:=GVL.V_220_T_1,pGK:=ADR(GVL.ZNV_HR_220_T_1),
    pGKH_1:=ADR(GVL.ZNH_HR_1_220_SV), pGKH_2:=ADR(GVL.ZNH_RTN_1_220),
    pActiv:=ADR(GVL.HR_220_T_1_act),pDet_vkl:=ADR(GVL.Isp_Mech[9].ON),
    pDet_otkl:=ADR(GVL.Isp_Mech[9].OFF), pDefect_circuit:=ADR(GVL.HR_220_T_1_CIR),
    pDefect_VBU:=ADR(GVL.HR_220_T_1_VBU));
GVL.HR_220_T_2.Init(Name:='ШИП 220 T-2',pHVCB:=GVL.V_220_T_2,pGK:=ADR(GVL.ZNV_HR_220_T_2),
    pGKH_1:=ADR(GVL.ZNH_HR_2_220_SV), pGKH_2:=ADR(GVL.ZNH_RTN_2_220),
    pActiv:=ADR(GVL.HR_220_T_2_act),pDet_vkl:=ADR(GVL.Isp_Mech[10].ON),
```

Продолжение Приложения Ж

pDet_otkl:=ADR(GVL.Isp_Mech[10].OFF), pDefect_circuit:=ADR(GVL.HR_220_T_2_CIR),
pDefect_VBU:=ADR(GVL.HR_220_T_2_VBU));
GVL.HR_1_220_SV.Init(Name:='ИП-1-220 CB',pHVCB:=GVL.SV_220,pGK:=ADR(GVL.ZNV_HR_1_220_SV),
pGKH_1:=ADR(GVL.ZNH_HR_1_220_SV), pGKH_2:=ADR(GVL.ZNH_RTN_1_220),
pActiv:=ADR(GVL.HR_1_220_SV_act), pDet_vkl:=ADR(GVL.Isp_Mech[11].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[11].OFF), pDefect_circuit:=ADR(GVL.HR_1_220_SV_CIR),
pDefect_VBU:=ADR(GVL.HR_1_220_SV_VBU));
GVL.HR_2_220_SV.Init(Name:='ИП-2-220 CB',pHVCB:=GVL.SV_220,pGK:=ADR(GVL.ZNV_HR_2_220_SV),
pGKH_1:=ADR(GVL.ZNH_HR_1_220_SV), pGKH_2:=ADR(GVL.ZNH_RTN_1_220),
pActiv:=ADR(GVL.HR_2_220_SV_act), pDet_vkl:=ADR(GVL.Isp_Mech[12].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[12].OFF), pDefect_circuit:=ADR(GVL.HR_2_220_SV_CIR),
pDefect_VBU:=ADR(GVL.HR_2_220_SV_VBU));
GVL.RTN_2_220.Init(Name:='PTH-2-220',pGKT:=ADR(GVL.ZNT_RTN_2_220),pGKH:=ADR(GVL.ZNH_RTN_2_220),
pActiv:=ADR(GVL.RTN_2_220_act),pTransformator:=ADR(GVL.Transf),
pDet_vkl:=ADR(GVL.Isp_Mech[13].ON),pDet_otkl:=ADR(GVL.Isp_Mech[13].OFF),
pDefect_circuit:=ADR(GVL.RTN_2_220_CIR), pDefect_VBU:=ADR(GVL.RTN_2_220_VBU));
GVL.RTN_1_220.Init(Name:='PTH-1-220',pGKT:=ADR(GVL.ZNT_RTN_1_220),pGKH:=ADR(GVL.ZNH_RTN_1_220),
pActiv:=ADR(GVL.RTN_1_220_act),pTransformator:=ADR(GVL.Transf),
pDet_vkl:=ADR(GVL.Isp_Mech[14].ON),pDet_otkl:=ADR(GVL.Isp_Mech[14].OFF),
pDefect_circuit:=ADR(GVL.RTN_1_220_CIR), pDefect_VBU:=ADR(GVL.RTN_1_220_VBU));
GVL.ZNH_HR_1_220_SV.Init(Name:='3ИИИ ИП-1-220 CB',pDC:=GVL.HR_1_220_SV,
pActiv:=ADR(GVL.ZNH_HR_1_220_SV_act), pDet_vkl:=ADR(GVL.Isp_Mech[15].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[15].OFF), pTransformator:=ADR(GVL.Transf),
pDefect_circuit:=ADR(GVL.ZNH_HR_1_220_SV_CIR));
GVL.ZNH_HR_2_220_SV.Init(Name:='3ИИИ ИП-2-220 CB',pDC:=GVL.HR_2_220_SV,
pActiv:=ADR(GVL.ZNH_HR_2_220_SV_act), pDet_vkl:=ADR(GVL.Isp_Mech[16].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[16].OFF), pTransformator:=ADR(GVL.Transf),
pDefect_circuit:=ADR(GVL.ZNH_HR_2_220_SV_CIR));
GVL.ZNL_LR_220_UKTALI.Init(Name:='3ИЛ ЛР 220 Юктали',pDC:=GVL.LR_220_UKTALI,
pActiv:=ADR(GVL.ZNL_LR_220_UKTALI_act), pDet_vkl:=ADR(GVL.Isp_Mech[17].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[17].OFF),pDefect_circuit:=ADR(GVL.ZNL_LR_220_UKTALI_CIR));
GVL.ZNV_LR_220_UKTALI.Init(Name:='3HB ЛР 220 Юктали',pDC:=GVL.LR_220_UKTALI,
pActiv:=ADR(GVL.ZNV_LR_220_UKTALI_act), pDet_vkl:=ADR(GVL.Isp_Mech[18].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[18].OFF), pDefect_circuit:=ADR(GVL.ZNV_LR_220_UKTALI_CIR));
GVL.ZNH_RTN_2_220.Init(Name:='3ИИИ PTH-2-220',pDC:=GVL.RTN_2_220,
pActiv:=ADR(GVL.ZNH_RTN_2_220_act), pDet_vkl:=ADR(GVL.Isp_Mech[19].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[19].OFF), pTransformator:=ADR(GVL.Transf),
pDefect_circuit:=ADR(GVL.ZNH_RTN_2_220_CIR));
GVL.ZNH_RTN_1_220.Init(Name:='3ИИИ PTH-1-220',pDC:=GVL.RTN_1_220,
pActiv:=ADR(GVL.ZNH_RTN_1_220_act), pDet_vkl:=ADR(GVL.Isp_Mech[20].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[20].OFF), pTransformator:=ADR(GVL.Transf),
pDefect_circuit:=ADR(GVL.ZNH_RTN_1_220_CIR));
GVL.ZNT_RTN_2_220.Init(Name:='3HT PTH-2-220',pDC:=GVL.RTN_2_220,
pActiv:=ADR(GVL.ZNT_RTN_2_220_act), pDet_vkl:=ADR(GVL.Isp_Mech[21].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[21].OFF), pDefect_circuit:=ADR(GVL.ZNT_RTN_2_220_CIR));
GVL.ZNT_RTN_1_220.Init(Name:='3HT PTH-1-220',pDC:=GVL.RTN_1_220,
pActiv:=ADR(GVL.ZNT_RTN_1_220_act), pDet_vkl:=ADR(GVL.Isp_Mech[22].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[22].OFF), pDefect_circuit:=ADR(GVL.ZNT_RTN_1_220_CIR));
GVL.ZNT_TR_220_T_2.Init(Name:='3HT TP 220 T-2',pDC:=GVL.TR_220_T_2,
pActiv:=ADR(GVL.ZNT_TR_220_T_2_act), pDet_vkl:=ADR(GVL.Isp_Mech[23].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[23].OFF), pDefect_circuit:=ADR(GVL.ZNT_TR_220_T_2_CIR));
GVL.ZNT_TR_220_T_1.Init(Name:='3HT TP 220 T-1',pDC:=GVL.TR_220_T_1,
pActiv:=ADR(GVL.ZNT_TR_220_T_1_act), pDet_vkl:=ADR(GVL.Isp_Mech[24].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[24].OFF), pDefect_circuit:=ADR(GVL.ZNT_TR_220_T_1_CIR));
GVL.ZNV_TR_220_T_1.Init(Name:='3HB TP 220 T-1',pDC:=GVL.TR_220_T_1,
pActiv:=ADR(GVL.ZNV_TR_220_T_1_act), pDet_vkl:=ADR(GVL.Isp_Mech[25].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[25].OFF), pDefect_circuit:=ADR(GVL.ZNV_TR_220_T_1_CIR));
GVL.ZNV_TR_220_T_2.Init(Name:='3HB TP 220 T-2',pDC:=GVL.TR_220_T_2,
pActiv:=ADR(GVL.ZNV_TR_220_T_2_act), pDet_vkl:=ADR(GVL.Isp_Mech[26].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[26].OFF), pDefect_circuit:=ADR(GVL.ZNV_TR_220_T_2_CIR));
GVL.ZNV_HR_220_T_2.Init(Name:='3HB ИП 220 T-2',pDC:=GVL.HR_220_T_2,
pActiv:=ADR(GVL.ZNV_HR_220_T_2_act), pDet_vkl:=ADR(GVL.Isp_Mech[27].ON),
pDet_otkl:=ADR(GVL.Isp_Mech[27].OFF), pDefect_circuit:=ADR(GVL.ZNV_HR_220_T_2_CIR));
GVL.ZNV_HR_220_T_1.Init(Name:='3HB ИП 220 T-1',pDC:=GVL.HR_220_T_1,
pActiv:=ADR(GVL.ZNV_HR_220_T_1_act), pDet_vkl:=ADR(GVL.Isp_Mech[28].ON),

Продолжение Приложения Ж

```

        pDet_otkl:=ADR(GVL.Isp_Mech[28].OFF), pDefect_circuit:=ADR(GVL.ZNV_HR_220_T_1_CIR));
GVL.ZNV_HR_2_220_SV.Init(Name:='3HB ИП-2-220 CB',pDC:=GVL.HR_2_220_SV,
        pActiv:=ADR(GVL.ZNV_HR_2_220_SV_act), pDet_vkl:=ADR(GVL.Isp_Mech[29].ON),
        pDet_otkl:=ADR(GVL.Isp_Mech[29].OFF), pDefect_circuit:=ADR(GVL.ZNV_HR_2_220_SV_CIR));
GVL.ZNV_HR_1_220_SV.Init(Name:='3HB ИП-1-220 CB',pDC:=GVL.HR_1_220_SV,
        pActiv:=ADR(GVL.ZNV_HR_1_220_SV_act), pDet_vkl:=ADR(GVL.Isp_Mech[30].ON),
        pDet_otkl:=ADR(GVL.Isp_Mech[30].OFF), pDefect_circuit:=ADR(GVL.ZNV_HR_1_220_SV_CIR));
GVL.ZNV_HR_220_UKTALI.Init(Name:='3HB ИП 220 Юктали',pDC:=GVL.HR_220_UKTALI,
        pActiv:=ADR(GVL.ZNV_HR_220_UKTALI_act), pDet_vkl:=ADR(GVL.Isp_Mech[31].ON),
        pDet_otkl:=ADR(GVL.Isp_Mech[31].OFF), pDefect_circuit:=ADR(GVL.ZNV_HR_220_UKTALI_CIR));

FUNCTION_BLOCK PlanExecutor
VAR_INPUT
    IN:BOOL;//переменная отвечающая за работу циклов поиска
    peMassiv_KA: ARRAY[1..31] OF DEVICE;//массив КА
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR

IF n>Card.nTask THEN
    IN:=FALSE;
    n:=1;
    i:=1;
    Done:=TRUE;
ELSE Done:=FALSE;
END_IF
IF IN THEN
    WHILE peMassiv_KA[i].Name <> Card.Tasks[n].Object DO
        i:=i+1;
    END_WHILE
    IF peMassiv_KA[i].Name = Card.Tasks[n].Object THEN
        IF Card.Tasks[n].WorkType = 'Включить' AND peMassiv_KA[i].State = 'off' THEN
            peMassiv_KA[i].TurnOn();
        ELSIF Card.Tasks[n].WorkType = 'Отключить' AND peMassiv_KA[i].State = 'on' THEN
            peMassiv_KA[i].TurnOff();
        ELSIF (Card.Tasks[n].WorkType = 'Отключить' AND peMassiv_KA[i].State = 'off') OR (Card.Tasks[n].WorkType =
'Включить' AND peMassiv_KA[i].State = 'on') THEN
            n:=n+1;
            i:=1;
        END_IF
    END_IF
END_IF

FUNCTION_BLOCK DC_LR IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:        STRING;
    myState:       STRING:='off';
    myMessage:     STRING:='';
    myHVCB_UKT:   POINTER TO HVCB_UKT;
    myGKL:         POINTER TO GK;
    myGKV:         POINTER TO GK;
    myActiv:       POINTER TO BOOL;
    myDet_vkl:     POINTER TO BOOL;
    myDet_otkl:    POINTER TO BOOL;
    myTransformer: POINTER TO BOOL;
    myDefect_circuit: POINTER TO BOOL;
    myDefect_VBU:  POINTER TO BOOL;
    timer:         TON;

```

Продолжение Приложения Ж

```
END_VAR

IF myState = 'off' THEN
  IF myDet_vk1^ THEN
    myState:= 'fault';
    myMessage:= 'Device error!';
  END_IF
ELSIF myState = 'GoOn' THEN
  IF myHVCB_UKT^.State='off' AND myGKL^.State='off' AND myGKV^.State='off' AND NOT myTransformer^
    AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_vk1^ THEN
      myState:= 'on';
      myMessage:= 'Done';
      timer(IN:=FALSE);
      myActiv^:=TRUE;
    END_IF
    IF timer.Q THEN
      myState:= 'fault';
      myMessage:= 'Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:= 'fault';
    IF myHVCB_UKT^.State='on' THEN
      myMessage:=CONCAT(myMessage,CONCAT(CONCAT(myHVCB_UKT^.Name,' '),myHVCB_UKT^.State));
    ELSIF myGKL^.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKL^.Name,' '),myGKL^.State));
    ELSIF myTransformer^=TRUE THEN myMessage:= CONCAT(myMessage, 'Присутствует напряжение на линии');
    ELSIF myGKV^.State='on' THEN
      myMessage:= CONCAT(myMessage, CONCAT(CONCAT(myGKV^.Name,' '),myGKV^.State));
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей обогрева ВБУ');
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'on' THEN
  IF myDet_otk1^ THEN
    myState:= 'fault';
    myMessage:= 'Device error!';
  END_IF
ELSIF myState = 'GoOff' THEN
  IF (myHVCB_UKT^.State='off') AND(NOT myTransformer^) AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otk1^ THEN
      myState:= 'off';
      myMessage:= 'Done';
      timer(IN:=FALSE);
      myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
      myState:= 'fault';
      myMessage:= 'Fault!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:= 'fault';
    IF myHVCB_UKT^.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB_UKT^.Name,' '),myHVCB_UKT^.State));
    ELSIF myTransformer^=TRUE THEN myMessage:= CONCAT(myMessage,' Присутствует напряжение на линии');
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей обогрева ВБУ');
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
END_IF
```

Продолжение Приложения Ж

```
ELSIF myState = 'fault' THEN
    myActiv^:=FALSE;
    GVL.Dialog_fault:=TRUE;
END_IF

outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
    Name:           STRING;
    pHVCB_UKT:      POINTER TO HVCB_UKT;
    pGKL:           POINTER TO GK;
    pGKV:           POINTER TO GK;
    pActiv:         POINTER TO BOOL;
    pDet_vkl:       POINTER TO BOOL;
    pDet_otkl:      POINTER TO BOOL;
    pTransformer:  POINTER TO BOOL;
    pDefect_circuit: POINTER TO BOOL;
    pDefect_VBU:   POINTER TO BOOL;
END_VAR

myName:=Name;
myHVCB_UKT:=pHVCB_UKT;
myGKL:=pGKL;
myGKV:=pGKV;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myTransformer:=pTransformer;
myDefect_circuit:=pDefect_circuit;
myDefect_VBU:=pDefect_VBU;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!'; //уже
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    IF (myHVCB_UKT^.State = 'on' OR myHVCB_UKT^.State = 'fault') AND myTransformer^ THEN
        command:= TO_STRING(myTransformer^);
        myMessage:= CONCAT(CONCAT(myHVCB_UKT^.Name, ' '),myHVCB_UKT^.State);
        myMessage:= CONCAT(myMessage, ' ');
        myMessage:= CONCAT(myMessage,command);
        Reset:=FALSE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        Reset:=TRUE;
    END_IF
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    IF (myHVCB_UKT^.State = 'on' OR myHVCB_UKT^.State = 'fault') AND myTransformer^ THEN
        IF myGKL^.State = 'on' OR myGKL^.State = 'fault' THEN
            myMessage:= CONCAT('Отключение невозможно т.к.!', ' ');
            myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB_UKT^.Name, ' '),myHVCB_UKT^.State));
        ELSIF myTransformer^=TRUE THEN
            myMessage:= 'Отключение невозможно т.к. присутствует напряжение на линии!';
        END_IF
    END_IF
END_IF
```


Продолжение Приложения Ж

```
END_IF
TurnOff:=FALSE;
GVL.Dialog_fault:=TRUE;
ELSE
    myMessage:='Attempt!'; //пытаемся
    myState:='GoOff'; // готов к отключению
    TurnOff:=TRUE;
    GVL.Dialog_fault:=FALSE;
END_IF
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF (myHVCB_UKT^.State='on' OR myHVCB_UKT^.State='fault') OR (myGKL^.State='on'
        OR myGKL^.State='fault') OR (myGKV^.State='on' OR myGKV^.State='fault') THEN
        IF myHVCB_UKT^.State='on' OR myHVCB_UKT^.State='fault' THEN
            myMessage:='Включение невозможно т.к. ';
            myMessage:=
                CONCAT(myMessage,CONCAT(CONCAT(myHVCB_UKT^.Name,'
'),myHVCB_UKT^.State));
        ELSIF myGKL^.State='on' OR myGKL^.State='fault' THEN
            myMessage:=' Включение невозможно т.к. ';
            myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKL^.Name,' '),myGKL^.State));
        ELSIF myGKV^.State='on' OR myGKV^.State='fault' THEN
            myMessage:=' Включение невозможно т.к. ';
            myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKV^.Name,' '),myGKV^.State));
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!'; // попытка
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK DC_T IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
```

Продолжение Приложения Ж

```
myState:          STRING:='off';
myMessage:        STRING:='';
myHVCB_T:         POINTER TO HVCB_T;
myGKT:            POINTER TO GK;
myGKV:            POINTER TO GK;
myActiv:          POINTER TO BOOL;
myDet_vkl:        POINTER TO BOOL;
myDet_otkl:       POINTER TO BOOL;
myDefect_circuit: POINTER TO BOOL;
myDefect_VBU      POINTER TO BOOL;
timer:            TON;

END_VAR

IF myState = 'off' THEN
  IF myDet_vkl^ THEN
    myState:='fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOn' THEN
  IF (myHVCB_T^.State='off') AND (myGKT^.State='off') AND (myGKV^.State='off') AND NOT myDefect_VBU^
  AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    myActiv^:=TRUE;
    IF myDet_vkl^ THEN
      myState:='on';
      myMessage:='Done';
      timer(IN:=FALSE);
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:='fault';
    IF myHVCB_T^.State='on' THEN
      myMessage:=CONCAT(myMessage,CONCAT(CONCAT(myHVCB_T^.Name,' '),myHVCB_T^.State));
    ELSIF myGKT^.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKT^.Name,' '),myGKT^.State));
    ELSIF myGKV^.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKV^.Name,' '),myGKV^.State));
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage, ' Неисправность цепей обогрева ВБУ');
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, ' Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'on' THEN
  IF myDet_otkl^ THEN
    myState:='fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOff' THEN
  IF myHVCB_T^.State='off' AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
      myState:='off';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Fault!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:='fault';
```

Продолжение Приложения Ж

```
IF myHVCB_T^.State='on' THEN
    myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB_T^.Name,' '),myHVCB_T^.State));
ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей обогрева ВБУ');
ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей питания и обогрева');
END_IF
timer(IN:=FALSE);
END_IF

ELSIF myState = 'fault' THEN
    myActiv^:=FALSE;
    GVL.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
    Name:          STRING;
    pHVCB_T:       POINTER TO HVCB_T;
    pGKT:          POINTER TO GK;
    pGKV:          POINTER TO GK;
    pActiv:        POINTER TO BOOL;
    pDet_vkl:      POINTER TO BOOL;
    pDet_otkl:     POINTER TO BOOL;
    pDefect_circuit: POINTER TO BOOL;
    pDefect_VBU:   POINTER TO BOOL;
END_VAR

myName:=Name;
myHVCB_T:=pHVCB_T;
myGKT:=pGKT;
myGKV:=pGKV;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myDefect_circuit:=pDefect_circuit;
myDefect_VBU:=pDefect_VBU;

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!'; TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF (myHVCB_T^.State = 'on' OR myHVCB_T^.State = 'fault') OR (myGKT^.State = 'on' OR myGKT^.State = 'fault') OR
        (myGKV^.State = 'on' OR myGKV^.State = 'fault') THEN
        IF myHVCB_T^.State = 'on' OR myHVCB_T^.State = 'fault' THEN
            myMessage:='Включение невозможно т.к. ';
            myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB_T^.Name,' '),myHVCB_T^.State));
        ELSIF myGKT^.State = 'on' OR myGKT^.State = 'fault' THEN
            myMessage:='Включение невозможно т.к. ';
            myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKT^.Name,' '),myGKT^.State));
        ELSIF myGKV^.State = 'on' OR myGKV^.State = 'fault' THEN
            myMessage:='Включение невозможно т.к. ';
            myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKV^.Name,' '),myGKV^.State));
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF
END_IF
```

Продолжение Приложения Ж

```
METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!';
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    IF myHVCB_T^.State = 'on' OR myHVCB_T^.State = 'fault' THEN
        myMessage:= CONCAT(CONCAT(myHVCB_T^.Name,' '),myHVCB_T^.State);
        Reset:=FALSE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        Reset:=TRUE;
    END_IF
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    IF myHVCB_T^.State = 'on' OR myHVCB_T^.State = 'fault' THEN
        myMessage:= CONCAT('Отключение невозможно т.к.!',');
        myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB_T^.Name,' '),myHVCB_T^.State));
        TurnOff:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        TurnOff:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK GK IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:='off';
    myMessage:       STRING:=' ';
    myDC:            Device;
    myActiv:         POINTER TO BOOL;
```

Продолжение Приложения Ж

```
myDet_vkl:      POINTER TO BOOL;
myDet_otkl:    POINTER TO BOOL;
myDefect_circuit: POINTER TO BOOL;
timer:TON;
END_VAR

IF myState = 'off' THEN
  IF myDet_vkl^ THEN
    myState:= 'fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOn' THEN
  IF myDC.State='off' AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_vkl^ THEN
      myState:='on';
      myMessage:='Done!';
      timer(IN:=FALSE);
      myActiv^:=TRUE;
    END_IF
    IF timer.Q THEN
      myState:= 'fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:= 'fault';
    IF myDC.State='on' THEN myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myDC.Name,' '),myDC.State));
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, ' Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'on' THEN
  IF myDet_otkl^ THEN
    myState:= 'fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOff' THEN
  IF NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
      myState:= 'off';
      myMessage:='Done!';
      timer(IN:=FALSE);
      myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
      myState:= 'fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:= 'fault';
    IF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, ' Неисправность цепей питания и обогрева');
  END_IF
  timer(IN:=FALSE);
END_IF
ELSIF myState = 'fault' THEN
  myActiv^:=FALSE;
  GVL.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
  Name:      STRING;
```

Продолжение Приложения Ж

```
pDC:           Device;
pActiv:        POINTER TO BOOL;
pDet_vkl:      POINTER TO BOOL;
pDet_otkl:     POINTER TO BOOL;
pDefect_circuit: POINTER TO BOOL;
END_VAR

myName:=Name;
myDC:=pDC;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myDefect_circuit:=pDefect_circuit;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!';
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    IF myDC.State = 'on' OR myDC.State = 'fault' THEN
        myMessage:= CONCAT(CONCAT(myDC.Name,' '),myDC.State);
        Reset:=FALSE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        Reset:=TRUE;
    END_IF
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    TurnOff:=TRUE;
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF myDC.State = 'on' OR myDC.State = 'fault' THEN
        myMessage:= CONCAT(CONCAT(myDC.Name,'-'),myDC.State);
        myMessage:=CONCAT('Включение невозможно т.к.',myMessage);
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF
```

Продолжение Приложения Ж

```
PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK GKH IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:='off';
    myMessage:       STRING:='';
    myDC:            Device;
    myActiv:         POINTER TO BOOL;
    myDet_vkl:       POINTER TO BOOL;
    myDet_otkl:      POINTER TO BOOL;
    myTransformer:   POINTER TO BOOL;
    myDefect_circuit: POINTER TO BOOL;
    timer:           TON;
END_VAR

IF myState = 'off' THEN
    IF myDet_vkl^ THEN
        myState:='fault';
        myMessage:='Device error!';
    END_IF
ELSIF myState = 'GoOn' THEN
    IF (myDC.State='off') AND (NOT myTransformer^) AND NOT myDefect_circuit^ THEN
        timer(IN:=TRUE, PT:=T#15S);
        IF myDet_vkl^ THEN
            myState:='on';
            myMessage:='Done!';
            timer(IN:=FALSE);
            myActiv^:=TRUE;
        END_IF
        IF timer.Q THEN
            myState:='fault';
            myMessage:='Device error!';
            timer(IN:=FALSE);
        END_IF
    ELSE
        myState:='fault';
        IF myDC.State='on' THEN myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myDC.Name,' '),myDC.State));
        ELSIF myTransformer^=TRUE THEN myMessage:= CONCAT(myMessage,' Присутствует напряжение на секции шин:');
        ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage,' Неисправность цепей питания и обогрева');
        END_IF
        timer(IN:=FALSE);
    END_IF
ELSIF myState = 'on' THEN
    IF myDet_otkl^ THEN
        myState:='fault';
        myMessage:='Device error!';
    END_IF
END_IF
```

Продолжение Приложения Ж

```
END_IF
ELSIF myState = 'GoOff' THEN
  IF NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
      myState:='off';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE myState:='fault';
    myMessage:= CONCAT(myMessage, 'Неисправность цепей питания и обогрева');
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'fault' THEN
  myActiv^:=FALSE;
  GVI.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
  Name:          STRING;
  pDC:           Device;
  pActiv:        POINTER TO BOOL;
  pDet_vkl:      POINTER TO BOOL;
  pDet_otkl:     POINTER TO BOOL;
  pTransformer:  POINTER TO BOOL;
  pDefect_circuit: POINTER TO BOOL;
END_VAR

myName:=Name;
myDC:=pDC;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myTransformer:=pTransformer;
myDefect_circuit:=pDefect_circuit;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
  myMessage:='Already!';
  Reset:=TRUE;
ELSIF myState = 'fault' THEN
  IF (myDC.State = 'on' OR myDC.State = 'fault') AND myTransformer^ THEN
    myMessage:= CONCAT(CONCAT(myDC.Name,' '),myDC.State);
    myMessage:= CONCAT(CONCAT(myMessage,''),'Напряжение на секции шин:');
    Reset:=FALSE;
  ELSE
    myMessage:='Attempt!';
    myState:='GoOff';
    Reset:=TRUE;
  END_IF
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
  myMessage:='Already!';
  TurnOff:=TRUE;
```


Продолжение Приложения Ж

```
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    TurnOff:=TRUE;
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF (myDC.State = 'on' OR myDC.State = 'fault') OR myTransformator^ THEN
        IF myDC.State = 'on' OR myDC.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myDC.Name,' '),myDC.State);
            myMessage:= CONCAT('Включение невозможно т.к.', myMessage);
        ELSIF myTransformator^=TRUE THEN
            myMessage:='Включение невозможно т.к. присутствует напряжение на секции шин';
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK HR IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:='off';
    myMessage:       STRING:='';
    myHVCB:          Device;
    myGK:             POINTER TO GK;
    myGKH_1:         POINTER TO GKH;
    myGKH_2:         POINTER TO GKH;
    myActiv:         POINTER TO BOOL;
```

Продолжение Приложения Ж

```
myDet_vkl:      POINTER TO BOOL;
myDet_otkl:     POINTER TO BOOL;
myDefect_circuit: POINTER TO BOOL;
myDefect_VBU   POINTER TO BOOL;
timer:         TON;

END_VAR

IF myState = 'off' THEN
  IF myDet_vkl^ THEN//положение разьед
    myState:='fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOn' THEN
  IF (myHVCB.State='off') AND (myGK^.State='off') AND (myGKH_1^.State='off') AND (myGKH_2^.State='off')
    AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_vkl^ THEN
      myState:='on';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=TRUE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:='fault';
    IF myHVCB.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB.Name,'-'),myHVCB.State));
    ELSIF myGK^.State='on' THEN myMessage:= CONCAT(myMessage, CONCAT(CONCAT(myGK^.Name,'-'),myGK^.State));
    ELSIF myGKH_1^.State='on' THEN
      myMessage:=CONCAT(myMessage, CONCAT(CONCAT(myGKH_1^.Name,'-'),myGKH_1^.State));
    ELSIF myGKH_2^.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKH_2^.Name,'-'),myGKH_2^.State));
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей обогрева ВБУ');
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'on' THEN
  IF myDet_otkl^ THEN
    myState:='fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOff' THEN
  IF myHVCB.State='off' AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
      myState:='off';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Fault!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:='fault';
    IF myHVCB.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myHVCB.Name,' '),myHVCB.State));
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage,' Неисправность цепей обогрева ВБУ');
```

Продолжение Приложения Ж

```
        ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage,' Неисправность цепей питания и обогрева');
        END_IF
        timer(IN:=FALSE);
    END_IF
ELSIF myState = 'fault' THEN
    myActiv^:=FALSE;
    GVL.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;
METHOD Init
VAR_INPUT
    Name:           STRING;
    pHVCB:          Device;
    pGK:             POINTER TO GK;
    pGKH_1:          POINTER TO GKH;
    pGKH_2:          POINTER TO GKH;
    pActiv:          POINTER TO BOOL;
    pDet_vkl:        POINTER TO BOOL;
    pDet_otkl:       POINTER TO BOOL;
    pDefect_circuit: POINTER TO BOOL;
    pDefect_VBU:     POINTER TO BOOL;
END_VAR

myName:=Name;
myHVCB:=pHVCB;
myGK:=pGK;
myGKH_1:=pGKH_1;
myGKH_2:=pGKH_2;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myDefect_circuit:=pDefect_circuit;
myDefect_VBU:=pDefect_VBU;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!'; //уже
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    IF (myHVCB.State = 'on' OR myHVCB.State = 'fault') AND (myGK^.State = 'on' OR myGK^.State = 'fault')
    AND (myGKH_1^.State = 'on' OR myGKH_1^.State = 'fault') AND (myGKH_2^.State = 'on' OR myGKH_2^.State = 'fault') THEN
        myMessage:= CONCAT(CONCAT(myHVCB.Name,' '),myHVCB.State);
        myMessage:= CONCAT(myMessage,' ');
        myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGK^.Name,' '),myGK^.State));
        myMessage:= CONCAT(myMessage,' ');
        myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKH_1^.Name,' '),myGKH_1^.State));
        myMessage:= CONCAT(myMessage,' ');
        myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKH_2^.Name,' '),myGKH_2^.State));
        Reset:=FALSE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        Reset:=TRUE;
    END_IF
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!'; //уже
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
```

Продолжение Приложения Ж

```
IF myHVCB.State = 'on' OR myHVCB.State = 'fault' THEN
    myMessage:= CONCAT(CONCAT(myHVCB.Name,' '),myHVCB.State);
    myMessage:=CONCAT('Невозможно отключить т.к.',myMessage);
    TurnOff:=FALSE;
    GVL.Dialog_fault:=TRUE;
ELSE
    myMessage:='Attempt!'; //пытаемся
    myState:='GoOff'; // готов к отключению
    TurnOff:=TRUE;
    GVL.Dialog_fault:=FALSE;
END_IF
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELIF myState = 'off' THEN
    IF (myHVCB.State = 'on' OR myHVCB.State = 'fault') AND (myGK^.State = 'on' OR myGK^.State = 'fault')
    AND (myGKH_1^.State = 'on' OR myGKH_1^.State = 'fault') AND (myGKH_2^.State = 'on' OR myGKH_2^.State = 'fault') THEN
        IF myHVCB.State = 'on' OR myHVCB.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myHVCB.Name,' '),myHVCB.State);
            myMessage:= CONCAT('Невозможно включить т.к.',myMessage);
        ELIF myGK^.State = 'on' OR myGK^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myGK^.Name,' '),myGK^.State);
            myMessage:= CONCAT('Невозможно включить т.к.',myMessage);
        ELIF myGKH_1^.State = 'on' OR myGKH_1^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myGKH_1^.Name,' '),myGKH_1^.State);
            myMessage:= CONCAT('Невозможно включить т.к.',myMessage);
        ELIF myGKH_2^.State = 'on' OR myGKH_2^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myGKH_2^.Name,' '),myGKH_2^.State);
            myMessage:= CONCAT('Невозможно включить т.к.',myMessage);
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!'; // попытка
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK HVCB_SV IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
```

Продолжение Приложения Ж

```
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:= 'off';
    myMessage:       STRING;
    myHR_SV_1:       POINTER TO HR;
    myHR_SV_2:       POINTER TO HR;
    myActiv:         POINTER TO BOOL;
    myDet_vkl:       POINTER TO BOOL;
    myDet_otkl:      POINTER TO BOOL;
    timer:           TON;
END_VAR

IF myState = 'off' THEN
    IF myDet_vkl^ THEN
        myState:= 'fault';
        myMessage:= 'Device error!';
    END_IF
ELSIF myState = 'GoOn' THEN
    IF (myHR_SV_1^.State='on') AND (myHR_SV_2^.State='on') THEN
        timer(IN:=TRUE, PT:=T#15S);
        IF myDet_vkl^ THEN
            myState:= 'on';
            myMessage:= 'Done!';
            timer(IN:=FALSE);
            myActiv^:=TRUE;
        END_IF
        IF timer.Q THEN
            myState:= 'fault';
            myMessage:= 'Device error!';
            timer(IN:=FALSE);
        END_IF
    ELSE
        myState:= 'fault';
        IF myHR_SV_1^.State='off' THEN myMessage:= CONCAT(CONCAT(myHR_SV_1^.Name, '),myHR_SV_1^.State);
        ELSIF myHR_SV_2^.State='off' THEN myMessage:= CONCAT(CONCAT(myHR_SV_2^.Name, '),myHR_SV_2^.State);
        END_IF
        timer(IN:=FALSE);
    END_IF
ELSIF myState = 'on' THEN
    IF myDet_otkl^ THEN
        myState:= 'fault';
        myMessage:= 'Device error!';
    END_IF
ELSIF myState = 'GoOff' THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
        myState:= 'off';
        myMessage:= 'Done!';
        timer(IN:=FALSE);
        myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
        myState:= 'fault';
        myMessage:= 'Device error!';
        timer(IN:=FALSE);
    END_IF
ELSIF myState = 'fault' THEN
    myActiv^:=FALSE;
    GVL.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;

METHOD Init
```

Продолжение Приложения Ж

```
VAR_INPUT
    Name:                STRING;
    pHR_SV_1:            POINTER TO HR;
    pHR_SV_2:            POINTER TO HR;
    pActiv:              POINTER TO BOOL;
    pDet_vkl:            POINTER TO BOOL;
    pDet_otkl:           POINTER TO BOOL;
END_VAR

myName:=Name;
myHR_SV_1:=pHR_SV_1;
myHR_SV_2:=pHR_SV_2;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!';
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    Reset:=TRUE;
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    TurnOff:=TRUE;
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF (myHR_SV_1^.State = 'off' OR myHR_SV_1^.State = 'fault') OR (myHR_SV_2^.State = 'off' OR myHR_SV_2^.State = 'fault') THEN
        IF myHR_SV_1^.State = 'off' OR myHR_SV_1^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myHR_SV_1^.Name,' '),myHR_SV_1^.State);
            myMessage:= CONCAT('Невозможно включить т.к.',myMessage);
        ELSIF myHR_SV_2^.State = 'off' OR myHR_SV_2^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myHR_SV_2^.Name,' '),myHR_SV_2^.State);
            myMessage:= CONCAT('Невозможно включить т.к.',myMessage);
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF
```

Продолжение Приложения Ж

```
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK HVCB_T IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:='off';
    myMessage:       STRING;
    myDC_T:          POINTER TO DC_T;
    myActiv:         POINTER TO BOOL;
    myDet_vkl:       POINTER TO BOOL;
    myDet_otkl:      POINTER TO BOOL;
    timer:           TON;
END_VAR

IF myState = 'off' THEN
    IF myDet_vkl^ THEN
        myState:='fault';
        myMessage:='Device error!';
    END_IF
ELSIF myState = 'GoOn' THEN
    IF myDC_T^.State='on' THEN
        timer(IN:=TRUE, PT:=T#15S);
        IF myDet_vkl^ THEN
            myState:='on';
            myMessage:='Done!';
            timer(IN:=FALSE);
            myActiv^:=TRUE;
        END_IF
        IF timer.Q THEN
            myState:='fault';
            myMessage:='Device error!';
            timer(IN:=FALSE);
        END_IF
    ELSE
        myState:='fault';
        myMessage:= CONCAT(CONCAT(myDC_T^.Name,' '),myDC_T^.State);
        timer(IN:=FALSE);
    END_IF
ELSIF myState = 'on' THEN
    IF myDet_otkl^ THEN
        myState:='fault';
        myMessage:='Device error!';
    END_IF
ELSIF myState = 'GoOff' THEN
    timer(IN:=TRUE, PT:=T#15S);
```

Продолжение Приложения Ж

```
IF myDet_otkl^ THEN
    myActiv^:=FALSE;
    myState:='off';
    myMessage:='Done!';
    timer(IN:=FALSE);
END_IF
IF timer.Q THEN
    myState:='fault';
    myMessage:='Device error!';
    timer(IN:=FALSE);
END_IF
ELSIF myState = 'fault' THEN
    myActiv^:=FALSE;
    GVL.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
    Name:           STRING;
    pDC_T:          POINTER TO DC_T;
    pActiv:         POINTER TO BOOL;
    pDet_vkl:       POINTER TO BOOL;
    pDet_otkl:      POINTER TO BOOL;
END_VAR

myName:=Name;
myDC_T:=pDC_T;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!';
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    Reset:=TRUE;
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';//уже
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    myMessage:='Attempt!'; //попытка
    myState:='GoOff';
    TurnOff:=TRUE;
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
```


Продолжение Приложения Ж

```
IF myDC_T^.State = 'off' OR myDC_T^.State = 'fault' THEN
    myMessage:= CONCAT(CONCAT(myDC_T^.Name,' '),myDC_T^.State);
    myMessage:= CONCAT('Включение невозможно т.к. ',myMessage);
    TurnOn:=FALSE;
    GVL.Dialog_fault:=TRUE;
ELSE
    myMessage:='Attempt!';
    myState:='GoOn';
    TurnOn:=TRUE;
    GVL.Dialog_fault:=FALSE;
END_IF
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK HVCB_UKT IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:= 'off';
    myMessage:       STRING;
    myHR_UKT:        POINTER TO HR;
    myDC_LR:         POINTER TO DC_LR;
    myActiv:         POINTER TO BOOL;
    myDet_vkl:       POINTER TO BOOL;
    myDet_otkl:      POINTER TO BOOL;
    timer:           TON;
END_VAR

IF myState = 'off' THEN
    IF myDet_vkl^ THEN
        myState:= 'fault';
        myMessage:= 'Device error!';
    END_IF
ELSIF myState = 'GoOn' THEN
    IF (myDC_LR^.State='on') AND (myHR_UKT^.State='on') THEN
        timer(IN:=TRUE, PT:=T#15S);
        IF myDet_vkl^ THEN
            myState:= 'on';
            myMessage:= 'Done!';
            timer(IN:=FALSE);
            myActiv^:=TRUE;
        END_IF
        IF timer.Q THEN
            myState:= 'fault';
            myMessage:= 'Device error!';
            timer(IN:=FALSE);
        END_IF
    END_IF
END_IF
```

Продолжение Приложения Ж

```
ELSE
    myState:= 'fault';
    IF myDC_LR^.State='off' THEN myMessage:= CONCAT(CONCAT(myDC_LR^.Name,' '),myDC_LR^.State);
    ELSIF myHR_UKT^.State='off' THEN myMessage:= CONCAT(CONCAT(myHR_UKT^.Name,' '),myHR_UKT^.State);
    END_IF
    timer(IN:=FALSE);
END_IF
ELSIF myState = 'on' THEN
    IF myDet_otkl^ THEN
        myState:= 'fault';
        myMessage:='Device error!';
    END_IF
ELSIF myState = 'GoOff' THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
        myState:= 'off';
        myMessage:='Done!';
        timer(IN:=FALSE);
        myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
        myState:= 'fault';
        myMessage:='Device error!';
        timer(IN:=FALSE);
    END_IF
ELSIF myState = 'fault' THEN
    myActiv^:=FALSE;
    GVL.Dialog_fault:=TRUE;
END_IF
outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
    Name:                STRING;
    pHR_UKT:              POINTER TO HR;
    pDC_LR:               POINTER TO DC_LR;
    pActiv:               POINTER TO BOOL;
    pDet_vkl:             POINTER TO BOOL;
    pDet_otkl:            POINTER TO BOOL;
END_VAR

myName:=Name;
myHR_UKT:=pHR_UKT;
myDC_LR:=pDC_LR;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!';
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    Reset:=TRUE;
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
```

Продолжение Приложения Ж

```
        TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    myMessage:='Attempt!';
    myState:='GoOff';
    TurnOff:=TRUE;
END_IF

METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF (myDC_LR^.State = 'off' OR myDC_LR^.State = 'fault') OR (myHR_UKT^.State = 'off' OR myHR_UKT^.State = 'fault') THEN
        IF myDC_LR^.State = 'off' OR myDC_LR^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myDC_LR^.Name, '),myDC_LR^.State);
            myMessage:= CONCAT('Включение невозможно т.к.',myMessage);
        ELSIF myHR_UKT^.State = 'off' OR myHR_UKT^.State = 'fault' THEN
            myMessage:= CONCAT(CONCAT(myHR_UKT^.Name, '),myHR_UKT^.State);
            myMessage:= CONCAT('Включение невозможно т.к.',myMessage);
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF

END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;

FUNCTION_BLOCK RTN IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName:          STRING;
    myState:         STRING:='off';
    myMessage:      STRING:='';
    myGKT:           POINTER TO GK;
    myGKH:           POINTER TO GK;
    myActiv:         POINTER TO BOOL;
    myDet_vkl:       POINTER TO BOOL;
    myDet_otkl:      POINTER TO BOOL;
    myTransformator: POINTER TO BOOL;
```

Продолжение Приложения Ж

```
myDefect_circuit:  POINTER TO BOOL;
myDefect_VBU      POINTER TO BOOL;
timer:            TON;
END_VAR

IF myState = 'off' THEN
  IF myDet_vkl^ THEN//положение разьед
    myState:='fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOn' THEN
  IF (myGKT^.State='off') AND (myGKH^.State='off') AND (myTransformer^:=FALSE)
    AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_vkl^ THEN
      myState:='on';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=TRUE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Device error!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:='fault';
    IF myGKT^.State='on' THEN myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKT^.Name,' '),myGKT^.State));
    ELSIF myGKH^.State='on' THEN
      myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKH^.Name,' '),myGKH^.State));
    ELSIF myTransformer^=TRUE THEN myMessage:= CONCAT(myMessage,'Присутствует напряжение на екции шин');
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage,'Неисправность цепей обогрева ВБУ');
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage,'Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'on' THEN
  IF myDet_otkl^ THEN
    myState:='fault';
    myMessage:='Device error!';
  END_IF
ELSIF myState = 'GoOff' THEN
  IF myTransformer^:=FALSE AND NOT myDefect_VBU^ AND NOT myDefect_circuit^ THEN
    timer(IN:=TRUE, PT:=T#15S);
    IF myDet_otkl^ THEN
      myState:='off';
      myMessage:='Done';
      timer(IN:=FALSE);
      myActiv^:=FALSE;
    END_IF
    IF timer.Q THEN
      myState:='fault';
      myMessage:='Fault!';
      timer(IN:=FALSE);
    END_IF
  ELSE
    myState:='fault';
    IF myTransformer^=TRUE THEN myMessage:= CONCAT(myMessage, 'Присутствует напряжение на линии');
    ELSIF myDefect_VBU^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей обогрева ВБУ');
    ELSIF myDefect_circuit^=TRUE THEN myMessage:= CONCAT(myMessage, 'Неисправность цепей питания и обогрева');
    END_IF
    timer(IN:=FALSE);
  END_IF
ELSIF myState = 'fault' THEN
  myActiv^:=FALSE;
```

Продолжение Приложения Ж

```
GVL.Dialog_fault:=TRUE;
END_IF

outline:=inline AND myActiv^;

METHOD Init
VAR_INPUT
    Name:                STRING;
    pGKH:                POINTER TO GK;
    pGKT:                POINTER TO GK;
    pActiv:              POINTER TO BOOL;
    pDet_vkl:            POINTER TO BOOL;
    pDet_otkl:           POINTER TO BOOL;
    pTransformer:       POINTER TO BOOL;
    pDefect_circuit:    POINTER TO BOOL;
    pDefect_VBU:        POINTER TO BOOL;
END_VAR

myName:=Name;
myGKH:=pGKH;
myGKT:=pGKT;
myActiv:=pActiv;
myDet_vkl:=pDet_vkl;
myDet_otkl:=pDet_otkl;
myTransformer:=pTransformer;
myDefect_circuit:=pDefect_circuit;
myDefect_VBU:=pDefect_VBU;

METHOD Reset : BOOL

IF myState = 'off' OR myState = 'on' THEN
    myMessage:='Already!'; //уже
    Reset:=TRUE;
ELSIF myState = 'fault' THEN
    IF (myGKT^.State = 'on' OR myGKT^.State = 'fault') AND (myGKH^.State = 'on' OR myGKH^.State = 'fault') THEN
        myMessage:= CONCAT(CONCAT(myGKT^.Name,' '),myGKT^.State);
        myMessage:= CONCAT(myMessage,' ');
        myMessage:= CONCAT(myMessage,CONCAT(CONCAT(myGKH^.Name,' '),myGKH^.State));
        myMessage:= CONCAT(myMessage,' ');
        myMessage:= CONCAT(CONCAT(myMessage,''),'Напряжение на секции шин:');
        Reset:=FALSE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        Reset:=TRUE;
    END_IF
END_IF

METHOD TurnOff : BOOL

IF myState = 'off' THEN
    myMessage:='Already!';
    TurnOff:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOff:=FALSE;
ELSIF myState = 'on' THEN
    IF myTransformer^ THEN
        myMessage:= 'Присутствует напряжение на секции шин';
        TurnOff:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!';
        myState:='GoOff';
        TurnOff:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF
```

Продолжение Приложения Ж

```
END_IF
END_IF

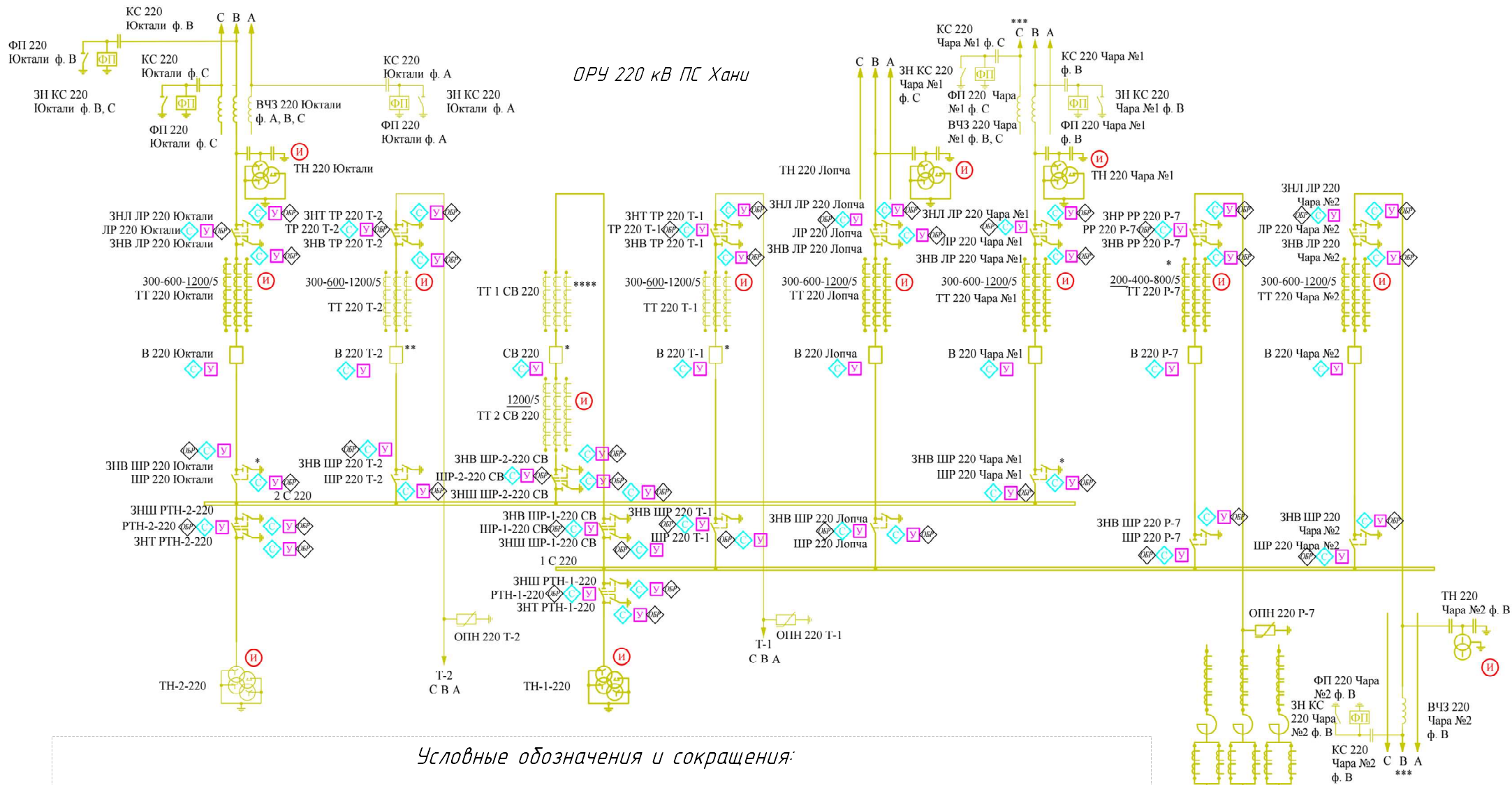
METHOD TurnOn : BOOL

IF myState = 'on' THEN
    myMessage:='Already!';
    TurnOn:=TRUE;
ELSIF myState = 'fault' THEN
    myMessage:='Fault!';
    TurnOn:=FALSE;
ELSIF myState = 'off' THEN
    IF (myGKT^.State = 'on' OR myGKT^.State = 'fault') OR (myGKH^.State = 'on' OR myGKH^.State = 'fault')THEN
        IF myGKT^.State = 'on' OR myGKT^.State = 'fault' THEN
            myMessage:=CONCAT(CONCAT(myGKT^.Name,' '),myGKT^.State);
            myMessage:=CONCAT('Включение невозможно т.к.',myMessage);
        ELSIF myGKH^.State = 'on' OR myGKH^.State = 'fault' THEN
            myMessage:=CONCAT(CONCAT(myGKH^.Name,' '),myGKH^.State);
            myMessage:=CONCAT('Включение невозможно т.к.',myMessage);
        END_IF
        TurnOn:=FALSE;
        GVL.Dialog_fault:=TRUE;
    ELSE
        myMessage:='Attempt!'; // попытка
        myState:='GoOn';
        TurnOn:=TRUE;
        GVL.Dialog_fault:=FALSE;
    END_IF
END_IF

PROPERTY Message : STRING
VAR
END_VAR
Message:=myMessage;

PROPERTY State : STRING
VAR
END_VAR
State:=MyState;

PROPERTY Name : STRING
VAR
END_VAR
Name:=myName;
```



ОРУ 220 кВ ПС Хани

Условные обозначения и сокращения:

- И - Объекты телемеханизации, телеизмерения
- С - Объект телемеханизации, телесигнализация
- У - Объект телемеханизации, телеуправление
- ОБР - Оперативная блокировка

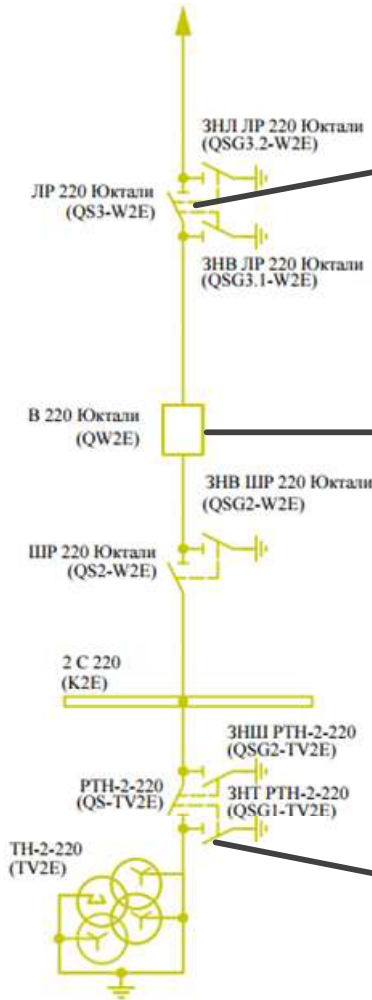
- ЗН - Заземляющий нож
- В - Элегазовый выключатель
- ЛР - Линейный разъединитель
- ШР - Шинный разъединитель
- СВ - Секционный выключатель
- РТН - Разъединитель трансформатора напряжения

- ТТ - Трансформатор тока
- ТН - Трансформатор напряжения
- ТР - Разъединитель понижающего трансформатора

				ВКР.194.003.15.03.04.СХ		
Изм	Лист	№ докум	Подп.	Дата	Однолинейная схема РУ 220 кВ ПС Хани	Лист 1
Состав	Масштаб	Масштаб	Масштаб	Масштаб	1:1	Масштаб
Проверка	Утверждение	Дата	Дата	Дата		
Разработчик	Проверка	Утверждение	Дата	Дата		
Выполнение	Проверка	Утверждение	Дата	Дата		
Дата	Дата	Дата	Дата	Дата		
Разработчик автоматизированной программы переключения для АСУ ТП подстанции Хани 220 кВ						Лист 1
						Листов 6
						АнГЧ зр. 941 об

Автоматизированная система управления процессами переключения

Ячейка №7
ВЛ 220 кВ Юктали – Хани с отпайкой на ПС Олёкма (W2E), ТН-2-220 (TV2E)



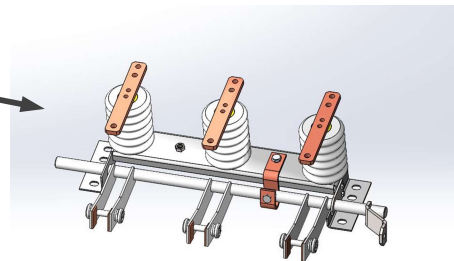
Разъединитель



Элегазовый выключатель



Земляющий нож



— АРМ оперативного персонала



— Контроллер среднего уровня ECU-4784



— Коммутатор управляемый сети Ethernet



— Терминал управления ЭКРА 242

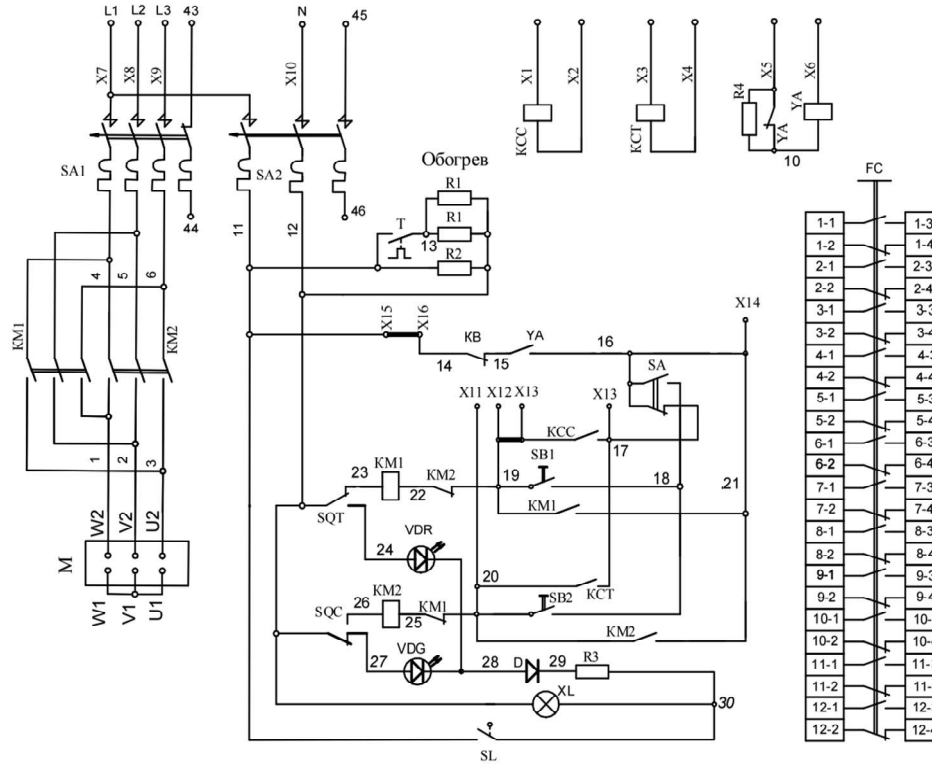


— Привод коммутационного аппарата

				ВКР.194.003.15.03.04.СХ		
Изм	Лист	№ докум	Подп.	Вариант	Дат.	Масштаб
Разраб.	Михайлов А.В.				У	1:1
Проектир.	Будылов А.А.					
Уточн.	Будылов А.А.				Лист 2	Листов 6
Разработчик: Автоматизированная программа переключения для АСУ ТП подстанции Хани 220 кВ				АнГЧ зр. 941 об		

Принципиальная схема привода разъединителя

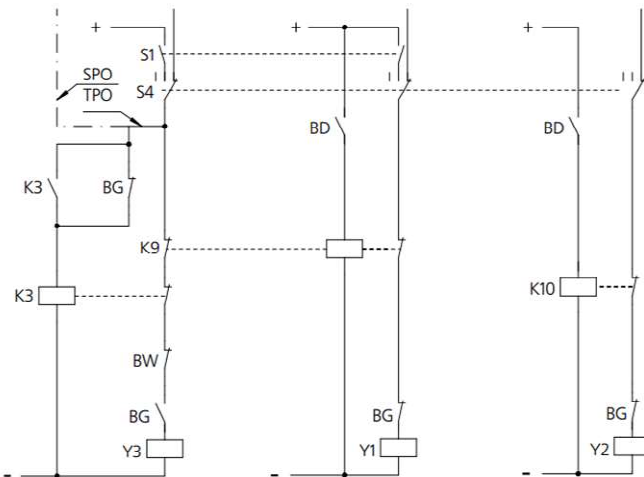
Положение на схеме – привод в отключенном состоянии, управление дистанционное, дверь закрыта.



Обозначение	Наименование
M	Электродвигатель
SA1, SA2	Выключатель автоматический
YA	Контактор длжировки
KM1, KM2	Контактор включения, отключения
SB1	Кнопка включения, черная
SB2	Кнопка отключения, красная
SQC, SQT	Конечные выключатели
SL	Выключатель подсветки и светодиодов
KB	Выключатель на рукоятке
SA	Переключатель дистанционное/местное
R1	Резистор 680 Ом
R2	Резистор 2 кОм
R3	Резистор 12 кОм
XL	Лампа подсветки
VDR	Светодиод зелёный
VDR	Светодиод красный
T	Датчик температуры
FC	Переключатель
KCC, KCT	Реле включения/отключения

Принципиальная схема привода выключателя

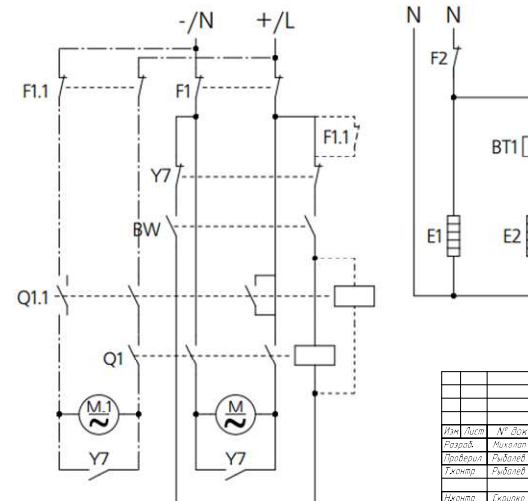
ВКЛЮЧЕНИЕ ОТКЛЮЧЕНИЕ 1 ОТКЛЮЧЕНИЕ 2



SPO – с полюсным управлением привода
 TPO – с трехполюсным управлением привода
 N – нейтраль
 L – под напряжением

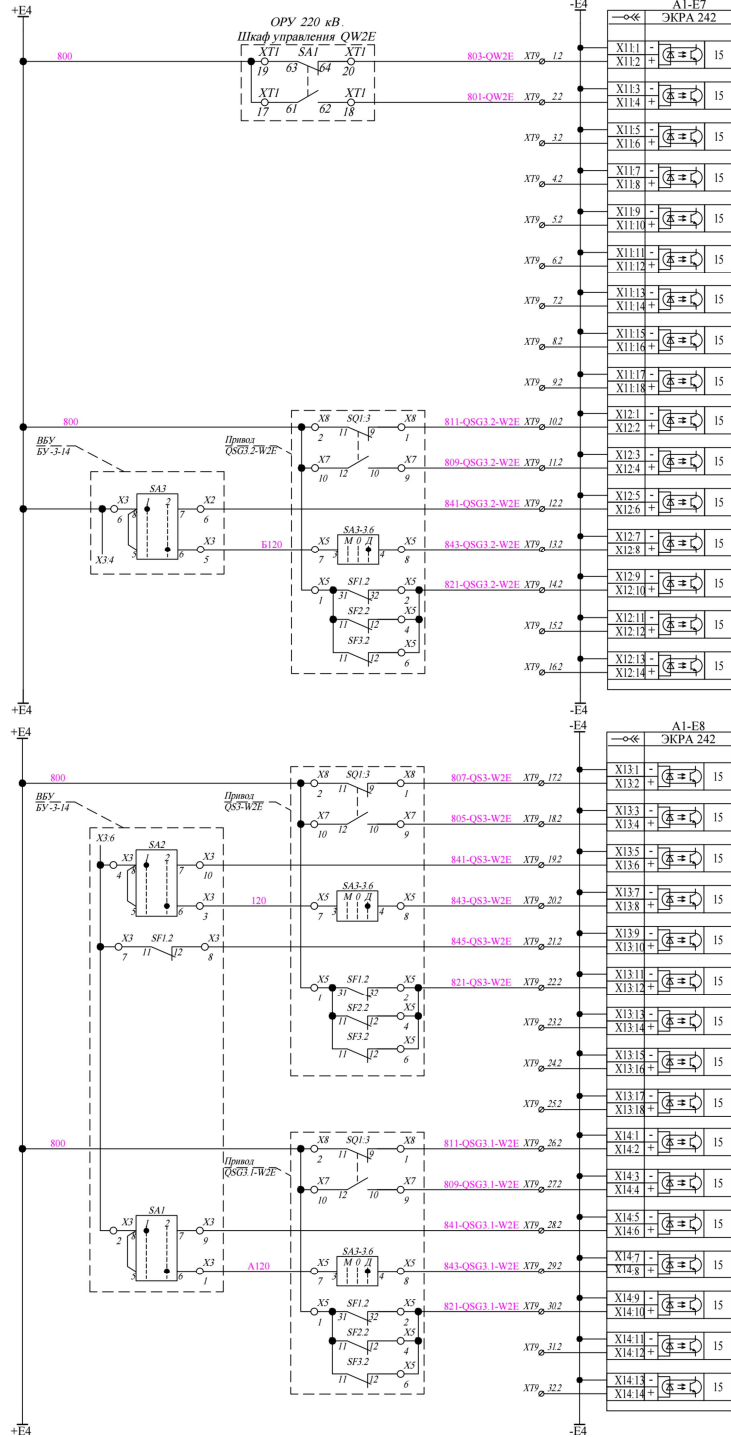
Принципиальная схема показана, когда выключатель находится в нормальном рабочем состоянии, т.е. элезаз под давлением, со взведенной пружинной включения, во включенном положении взвода и с многопозиционным переключателем выбора режима в положении «Дистанционное».

ЭЛЕКТРОДВИГАТЕЛЬ НАГРЕВАТЕЛЬ



ВКР.194.003.15.03.04.СХ				Лист	Масштаб	
Изм.	Лист	№ докум.	Подп.	Дата	1:1	
Разработчик	Михайлов А.С.	Проверщик	Будышев А.А.	Уд.		
Утвержден	Будышев А.А.				Лист 3	Листов 6
Выполнено	Скворцова Д.В.	Разработка автоматизированной программы переключения для АСУ ТП подстанции Хаши 220 кВ			АнГЧ зр. 941 об	
Уд.	Скворцова Д.В.					

Схемы подключения к входу ЭКРА 242



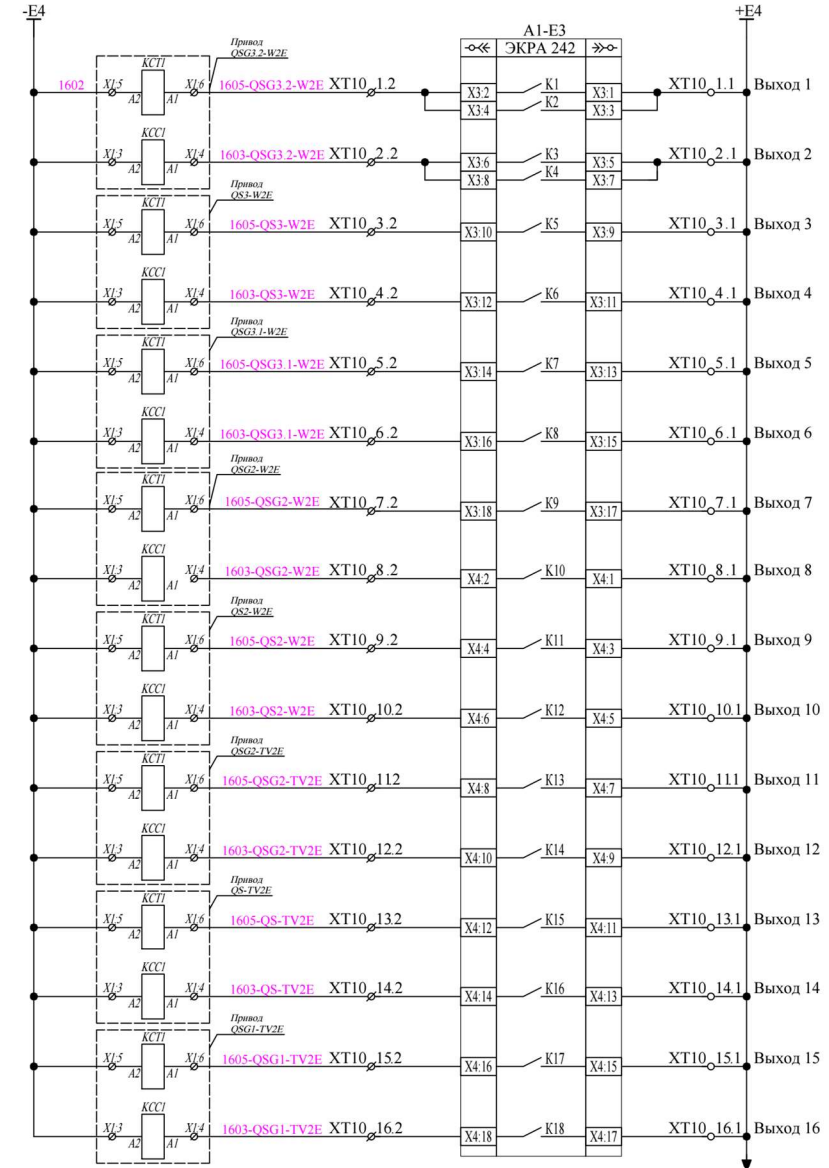
Положение выключателя "Отключен"	В 220 Юкгаин
Положение выключателя "Включен"	

Положение - "Отключен"	3ИИЛ ПР 220 Юкгаин
Положение - "Включен"	
Режим управления - "Местное"	
Режим управления - "Дистанционное"	
Неисправность цепей питания и оборота	

Положение - "Отключен"	ЛР 220 Юкгаин
Положение - "Включен"	
Режим управления - "Местное"	
Режим управления - "Дистанционное"	
Неисправность цепей оборота ВБУ	
Неисправность цепей питания и оборота	

Положение - "Отключен"	3ИВ ПР 220 Юкгаин
Положение - "Включен"	
Режим управления - "Местное"	
Режим управления - "Дистанционное"	
Неисправность цепей питания и оборота	

Схемы подключения выходов ЭКРА 242



Команда - "Отключить"	3ИИЛ ПР 220 Юкгаин
Команда - "Включить"	
Команда - "Отключить"	ЛР 220 Юкгаин
Команда - "Включить"	
Команда - "Отключить"	3ИВ ЛР 220 Юкгаин
Команда - "Включить"	
Команда - "Отключить"	3ИВ ПР 220 Юкгаин
Команда - "Включить"	
Команда - "Отключить"	ПР 220 Юкгаин
Команда - "Включить"	
Команда - "Отключить"	3ИИЛ РПН-2-220
Команда - "Включить"	
Команда - "Отключить"	РПН-2-220
Команда - "Включить"	
Команда - "Отключить"	3ИЛ РПН-2-220
Команда - "Включить"	
Команда - "Отключить"	РПН-2-220
Команда - "Включить"	
Команда - "Отключить"	3ИЛ РПН-2-220
Команда - "Включить"	

Подсистема формирования и хранения плана переключений

Заголовок: АПП для включения в работу 7 Ячейки

Ячейка	Объект	Операция	Дата и время	Статус
ВЛ 220кВ Юктали-Хани	ШР 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.
ВЛ 220кВ Юктали-Хани	ЛР 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.
ВЛ 220кВ Юктали-Хани	ШР 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.
ВЛ 220кВ Юктали-Хани	В 220 Юктали	Включить	DT#2023-01-01-00:00:00	Актив.

Информация:

Добавить задание Удалить задание Записать в файл Прочитать из файла

Выполнить

Окно формирования ошибок цепей обогрева ВБУ

Список КА:

- ЛР 220 Юктали
- ШР 220 Юктали
- РТН-2-220
- ТР 220 Т-2
- ЗНВ ШР 220 Т-2
- ШР 220 Т-2
- ШР-1-220 СВ
- ЗНВ ШР-2-220 СВ
- ШР-2-220 СВ
- ЗНШ ШР-2-220 СВ
- РТН-1-220
- ТР 220 Т-1
- ШР 220 Т-1

Закреть

Окно ошибки

ОШИБКА!

Коммутационный аппарат:

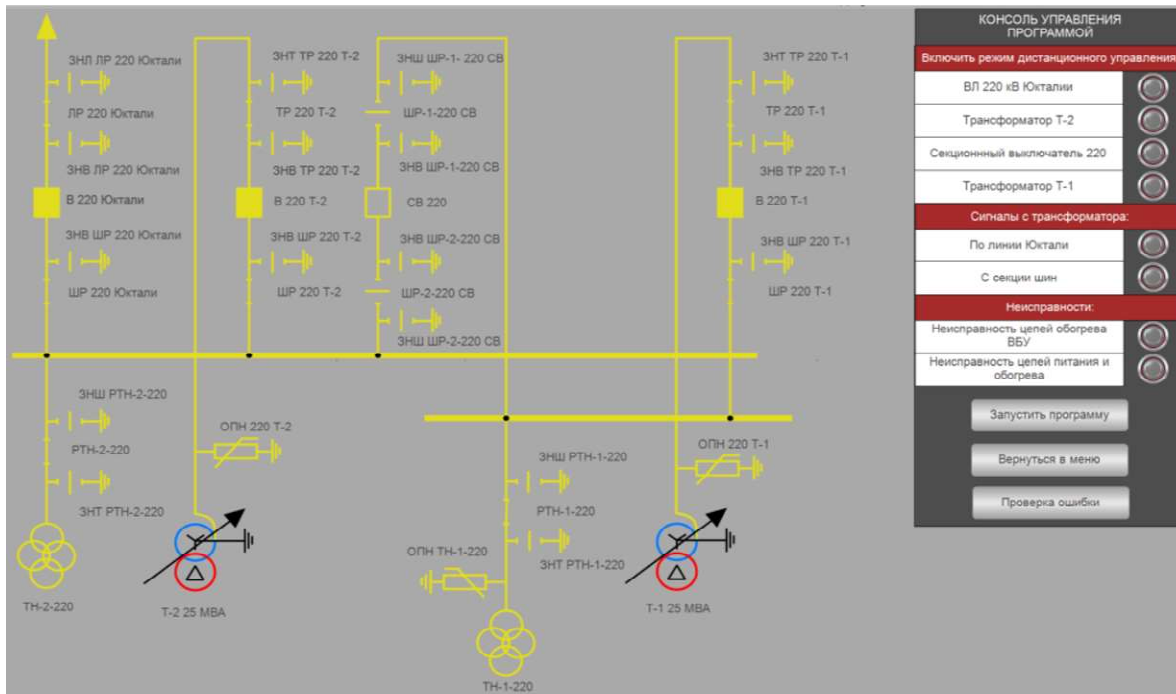
ЛР 220 Юктали

Причина ошибки:

Включение невозможно т.к. ЗНЛ ЛР 220 Юктали on

Закреть

Подсистема выполнения плана переключений

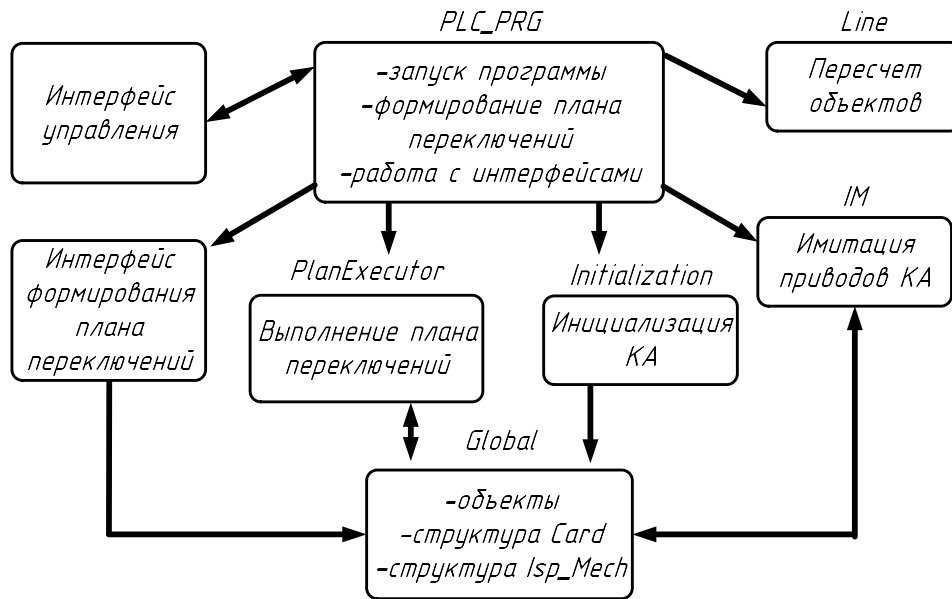


Окно формирования ошибок цепей питания и обогрева

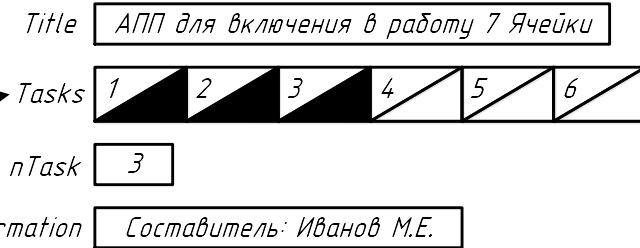
Список коммутационных аппаратов по ячейкам:

ВЛ 220 кВ Юктали:	Трансформатор Т-2:	Секционный выключатель 220:
ЗНЛ ЛР 220 Юктали	ЗНТ ТР 220 Т-2	ЗНШ ШР-1-220 СВ
ЛР 220 Юктали	ТР 220 Т-2	ШР-1-220 СВ
ЗНВ ЛР 220 Юктали	ЗНВ ТР 220 Т-2	ЗНВ ШР-1-220 СВ
ЗНВ ШР 220 Юктали	ЗНВ ШР 220 Т-2	ЗНВ ШР-2-220 СВ
ШР 220 Юктали	ШР 220 Т-2	ШР-2-220 СВ
ЗНШ РТН-2-220		ЗНШ ШР-2-220 СВ
РТН-2-220		ЗНШ РТН-1-220
ЗНТ РТН-2-220		РТН-1-220
		ЗНТ РТН-1-220
	Трансформатор Т-1:	
	ЗНТ ТР 220 Т-1	
	ТР 220 Т-1	
	ЗНВ ТР 220 Т-1	
	ЗНВ ШР 220 Т-1	
	ШР 220 Т-1	

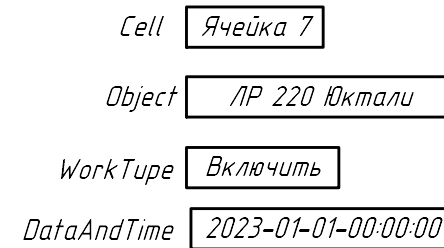
Закреть



Структура WorkCard



Структура Task



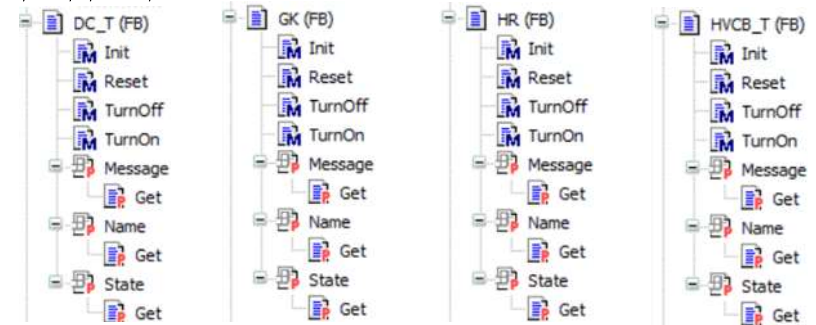
Объектно-ориентированное программирование

Разъединители трансформаторов

Заземляющие ножи

Шинные разъединители

Выключатель трансформаторов

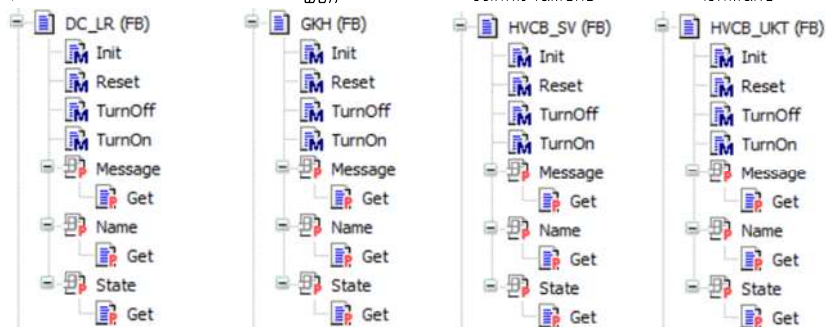


Линейный разъединитель

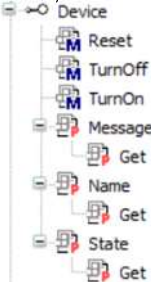
Заземляющие ножи шин

Секционный выключатель

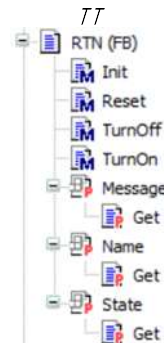
Выключатель линии Юкталу



Интерфейс - «Устройство»



Разъединители ТТ



Список локальных переменных функционального блока DC_T

```

FUNCTION_BLOCK DC_T IMPLEMENTS Device
VAR_INPUT
    inline:BOOL;
END_VAR
VAR_OUTPUT
    outline:BOOL;
END_VAR
VAR
    myName: STRING;
    myState: STRING:='off';
    myMessage: STRING:='';
    myHVCB_T, myGKT, myGKV: DEVICE
    myActiv, myDet_vkl, myDet_otkl: POINTER TO BOOL;
    myDefect_circuit, myDefect_VBU: POINTER TO BOOL;
    timer:TONT;
END_VAR
    
```