

УТВЕРЖДАЮ

Директор

ООО «ДЕЦИМА»

_____ А.А. Шкляев

“ _____ ” _____ 2018

Программный комплекс КОТМИ-14

WEB-доступ

Руководство системного программиста

Лист утверждения

ЯКШГ.00067-01 91 01-05 32 - ЛУ

Руководитель разработки

Начальник лаборатории

_____ А.В. Тумаков

“ _____ ” _____ 2018

Ответственный исполнитель

Ведущий инженер-программист

_____ М.Ю. Дьяченко

“ _____ ” _____ 2018

Инв.№ подл.	
Подп. и дата	
Взам. инв. №	
Инв.№ дубл.	
Подп. и дата	

Утвержден
ЯКШГ.00067-01 91 01-05 32 - ЛУ

ПРОГРАММНЫЙ КОМПЛЕКС КОТМИ-14

WEB-ДОСТУП

Руководство системного программиста

ЯКШГ.00067-01 91 01-05 32

Листов 37

Инв.№ подл.	Подп. и дата	Взам. инв. №	Инв.№ дубл.	Подп. и дата

АННОТАЦИЯ

Данный документ является руководством системного программиста (администратора) для настройки и эксплуатации программного обеспечения (ПО) по организации удаленного доступа к данным сервера приложений «КОТМИ-14» через WEB (далее по тексту – WEB-доступ) программного комплекса (ПК) КОТМИ-14.

В документе дается назначение и условия выполнения программы. Приводятся необходимые сведения по использованию функциональных модулей программы на этапе эксплуатации.

СОДЕРЖАНИЕ

Лист

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ	5
1.1. ОБОЗНАЧЕНИЕ И НАИМЕНОВАНИЕ ПРОГРАММЫ	5
1.2. НАЗНАЧЕНИЕ И ФУНКЦИИ ПРОГРАММЫ	5
1.3. НЕОБХОДИМЫЕ ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА	5
2. СТРУКТУРА И АЛГОРИТМ РАБОТЫ ПО «WEB-ДОСТУП»	7
2.1. СОСТАВНЫЕ ЧАСТИ ПО «WEB-ДОСТУП» И СВЯЗИ МЕЖДУ НИМИ	7
2.2. АЛГОРИТМ РАБОТЫ ПРОГРАММЫ	7
2.3. СТРУКТУРА ПРОГРАММЫ HTTP-ШЛЮЗ	9
2.4. СТРУКТУРА КЛИЕНТСКОЙ ЧАСТИ ПО «WEB-ДОСТУП»	10
2.5. SVG-БИБЛИОТЕКА	10
2.6. СВЯЗЬ С ДРУГИМИ ПРОГРАММАМИ	12
2.7. ИСПОЛЬЗУЕМЫЕ МЕТОДЫ	12
3. НАСТРОЙКА ПРОГРАММЫ	13
3.1. СОСТАВ ДИСТРИБУТИВА	13
3.2. УСТАНОВКА ПРОГРАММЫ	14
3.3. СТРУКТУРА КОНФИГУРАЦИОННОГО ФАЙЛА ПРОГРАММЫ HTTP-ШЛЮЗА	14
3.4. АРГУМЕНТЫ КОМАНДНОЙ СТРОКИ	15
3.5. ПОДКЛЮЧЕНИЕ ЧЕРЕЗ WEB-БРАУЗЕР К ПК КОТМИ-14	16
4. НАСТРОЙКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА	18
5. ДОБАВЛЕНИЕ ЭЛЕМЕНТОВ В БИБЛИОТЕКУ SVG	19
6. ПОДГОТОВКА СХЕМ ДЛЯ ОТОБРАЖЕНИЯ В WEB-БРАУЗЕРЕ	27
7. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	34
8. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ	35

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1. Обозначение и наименование программы

Наименование: программное обеспечение (ПО) по организации удаленного доступа к данным сервера приложений программного комплекса «КОТМИ-14» через WEB.

Обозначение: «WEB-доступ».

1.2. Назначение и функции программы

ПО «WEB-доступ» обеспечивает взаимодействие пользователя с сервером приложений ПК КОТМИ-14 и предоставляет ему функциональность в соответствии с требованиями решаемых задач. ПО «WEB-доступ» предназначено для отображения режимной и служебной информации.

Основными задачами, решаемыми ПО «WEB-доступ», являются:

- визуальное представление различных видов системной информации на экране пользователя в окне браузера. Информация может быть представлена в виде схем, отчетов, графиков, документов.

Соответственно задачам ПО «WEB-доступ» предназначено для выполнения следующих функций:

- Просмотра форм отображения ПК КОТМИ-14 (схем, документов, диаграмм) в браузере.
- Отображение событий ПК КОТМИ-14 в браузере.

1.3. Необходимые технические и программные средства

Серверная программа ПО «WEB-доступ» работает в стандартном персональном компьютере под управлением операционной системы Windows. Используются стандартные сервисы, предоставляемые этой операционной системой.

Для клиентской части необходимо устройство с любой операционной системой. Главным критерием является наличие WEB-браузера.

Для использования серверной программы ПО «WEB-доступ» рекомендуются следующие минимальные аппаратные требования к ПК:

- Оперативная память — 4 ГБ.
- Дисковое пространство — 1 ГБ.
- Экран размером не менее 1280 x 1024 пикселей.
- Наличие манипулятора «мышь» с интерфейсом PS/2 или USB.
- Наличие 101/102-кнопочной клавиатуры с русской и латинской раскладкой.
- Скорость канала связи Ethernet с сервером системы — 10 Мбит/сек.

Для использования клиентской части достаточно наличия стационарного или портативного устройства (компьютер, планшет, телефон) с любой операционной системы, единственным условием является подключение к сети и наличие веб-браузера.

Скорость работы ПО «WEB-доступ» на конкретном устройстве зависит также от отдельных его комплектующих (процессора, оперативной памяти и др.).

Серверная программа ПО «WEB-доступ» функционирует в режиме постоянного взаимодействия с сервером приложений ПК КОТМИ-14. Поэтому, для ее корректной работы важно наличие надежного и быстрого канала связи. Скорость канала меньше 10 Мбит/сек, может приводить временным к задержкам в работе программы. Потеря связи приводит к останову и закрытию программы.

2. СТРУКТУРА И АЛГОРИТМ РАБОТЫ ПО «WEB-ДОСТУП»

2.1. Составные части ПО «WEB-доступ» и связи между ними

Программа WEB-доступ состоит из следующих составных частей:

- HTTP-шлюз;
- клиентское приложение в WEB-браузере.

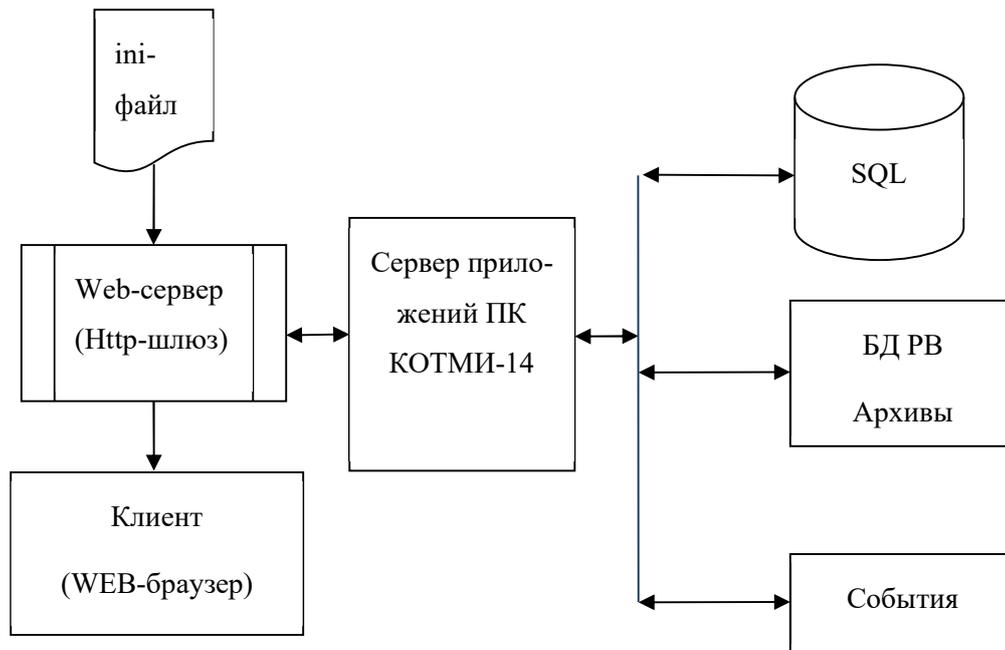


Рисунок 2.1 - Структура ПО «WEB-доступ»

2.2. Алгоритм работы программы

Программа HTTP-шлюз может быть запущена на выполнение, как из командной строки, так и программно. Программа читает ini-файл, из конфигурации берет адрес сервера приложений ПК КОТМИ-14 и прочие настройки, необходимые для дальнейшей работы. Если адрес сервера приложений не задан явно в ini-файле, то используется значение по умолчанию: LocalHost:1312.

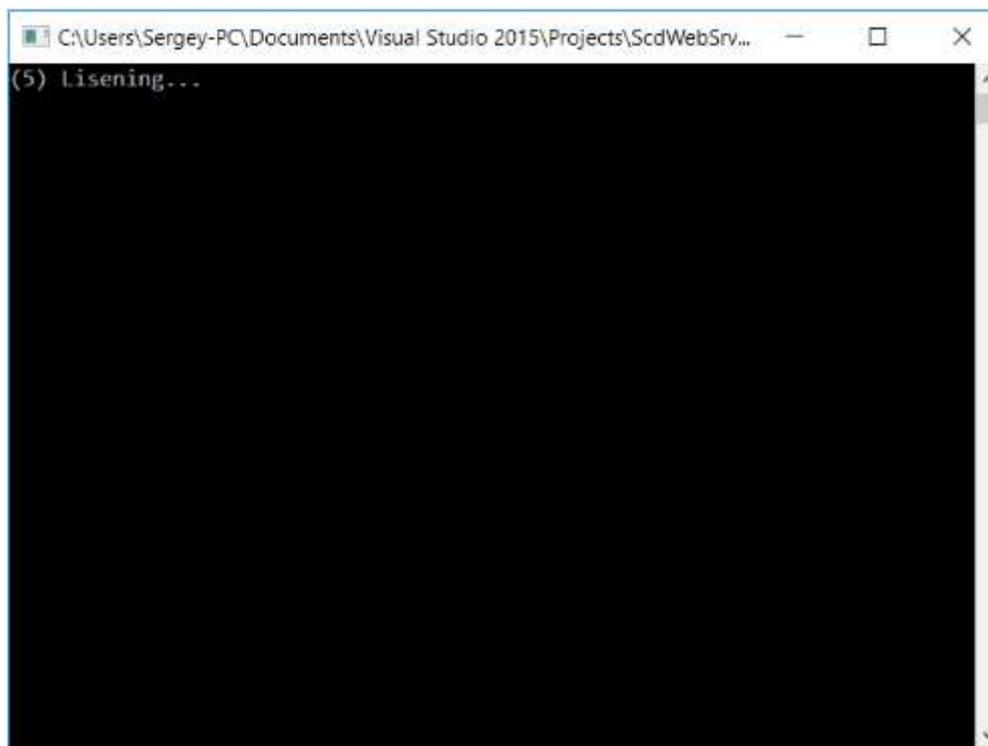


Рисунок 2.2 - Консольное окно HTTP-шлюза

После запуска программы HTTP-шлюза, пользователь может с помощью браузера перейти по адресу, на котором запущен HTTP-шлюз. Начальной страницей является html-страница с названием «auth.html», на которой располагается форма входа в систему.

A screenshot of a web form for authentication. It features two input fields: one labeled "Логин" (Login) containing the text "user", and another labeled "Пароль" (Password) containing the text "password". Below these fields is a blue button with the text "Отправить" (Send).

Рисунок 2.3 - Форма входа в систему

В форме пользователь указывает свой логин и пароль.

После успешной авторизации, пользователь может с помощью основного меню, открывать в браузере различные формы отображения. В зависимости от категории и прав пользователя, набор доступных форм отображения может различаться.

2.3. Структура программы HTTP-шлюз

Все HTTP-запросы обрабатываются в функции `CWebServer::onReadyRead()`.

Запросы с заголовком GET используются для получения html-страниц, файлов js, css, jpeg, svg.

Запросы с заголовком POST используются для всех остальных случаев.

Запросы POST имеют структуру заголовка и данных в формате JSON, в котором описывается тип запроса с помощью атрибута «**query**», «**GUID**» - идентификатор пользователя и различные дополнительные данные.

Типы обрабатываемых запросов (указанных в атрибуте **query**):

- «**login**». Вход пользователя в систему. Дополнительные атрибуты «**user**» и «**password**»;
- «**getmenu**». Получение дерева меню;
- «**geticon**». Получение иконок из сервера приложений ПК КОТМИ-14 и сохранение их в кэш HTTP-шлюза. Дополнительный атрибут «**img**». Допускается получение всех иконок одним запросом, указав вместо имени *;
- «**getsound**». Получение звуков иконок из сервера приложений ПК КОТМИ-14 и сохранение их в кэш HTTP-шлюза. Дополнительный атрибут «**snd**». Допускается получение всех звуков одним запросом указав вместо имени *;
- «**doDocOpen**». Запрос на открытие документа. Дополнительный атрибут «**docName**»;
- «**doDocTimeArgs**». Запрос на получение данных документа за промежуток времени. Дополнительные атрибуты «**docGuid**» - id документа, «**Time**» - время документа, «**reqTimeDepth**» - глубина запроса данных;
- «**doDocTimeNow**». Запрос на получение данных документа на текущее время. Дополнительные атрибуты «**docGuid**» - id документа, «**reqTimeDepth**» - глубина запроса данных;
- «**doDocTime**». Запрос на получение данных документа на текущее время на заданную глубину и подписка документа. Дополнительные атрибуты «**docGuid**» - id документа, «**reqTimeDepth**» - глубина запроса данных;
- «**doDocUpdate**». Получение данных документа по подписке. Дополнительный атрибут «**docGuid**» - id документа;

- «doMapUpdate». Получение данных для Яндекс карт. Дополнительный атрибут «guidList» – список идентификаторов линий;
- «getCustomers». Получение данных по потребителям. Дополнительный атрибут «keyLink» – id оборудования, для которого необходимо вывести список потребителей;
- «doPassport». Получение паспорта оборудования или паспорта измерения. Дополнительный атрибут «rtid» – id оборудования или измерения;
- «doDocClose». Запрос на закрытие документа. Дополнительный атрибут «docGuid» - id документа.

2.4. Структура клиентской части ПО «WEB-доступ»

Главным объектом является массив вкладок InsetsCharts[].

Он хранит в себе «GUID» открытого документа, тип документа, «id» вкладки с контентом и другие атрибуты в зависимости от типа документа. После выбора документа из дерева меню, посылается запрос в HTTP-шлюз на открытие документа и запрос на получение данных по подписке. В зависимости от типа документа строится «menuBar» с различными кнопками для взаимодействия с отображением документа.

2.5. SVG-библиотека

Для отображения SVG-схем используется библиотека элементов. Библиотека состоит из двух файлов:

1. SVG-файл с графическим представлением элементов;
2. JavaScript (Js)-файл для анимации элементов в зависимости от приходящих данных с сервера.

SVG-файл библиотеки включает в себя блок «defs», в котором описаны различные типы штриховки (применяются для стилей). Далее идёт графическое описание различных типов элементов.

Например, описание элемента «Разъединитель»:

```

<!--Разъединитель Disconnecter-->
<g class="disconnector" type="162">
  <!--Шаблон разъединителя-->
  <symbol id="disconnector_simple" preserveAspectRatio="xMidYMid meet">
    <rect class="substrate invisible" x="1" y="1" width="16" height="23"/>
    <!--Временное добавление для проверки отображения-->
    <line x1="5" y1="4" x2="12" y2="4"/>
    <line x1="8.5" y1="20" x2="8.5" y2="4"/>
    <line x1="5" y1="20" x2="12" y2="20"/>
  </symbol>

  <!--Замкнутый разъединитель-->
  <symbol id="disconnector_on" preserveAspectRatio="xMidYMid meet">
    <rect x="1" y="1" width="16" height="23" class="substrate invisible"/>
    <line x1="5" y1="4" x2="12" y2="4"/>
    <line x1="8.5" y1="20" x2="8.5" y2="4"/>
    <line x1="5" y1="20" x2="12" y2="20"/>
  </symbol>

  <!--Разомкнутый разъединитель-->
  <symbol id="disconnector_off" preserveAspectRatio="xMidYMid meet">
    <rect x="1" y="1" width="15" height="23" class="substrate invisible"/>
    <line x1="5" y1="4" x2="12" y2="4"/>
    <ellipse cx="8.5" cy="19.5" rx="1.5" ry="1.5"/>
    <line x1="9" y1="18" x2="14" y2="8"/>
  </symbol>
</g>

```

Рисунок 2.4 – Пример элемента схемы в формате SVG

Атрибут «**type**» в теге «**g**» является обязательным для определения размера и имени элемента. В тег «**g**» помещаются теги «**symbol**», в которых указываются графические примитивы элемента. Первым (название элемента + `_simple`) указываются общие графические примитивы, которые совпадают для всех состояний элемента. В каждом теге «**symbol**» первым графическим примитивом указывается прямоугольник описывающий элемент. Для этого прямоугольника необходимо указать атрибут «**class="substrate invisible"**». По размерам этого прямоугольника будут центрироваться элементы на схеме при конвертации.

Js-файл (`testClasses.js`) описывает смену изображения при различных состояниях элемента. В данном файле реализуются все типы элементов в виде классов. Родительский класс «**Element**» – в нём реализован общий функционал для всех элементов. Далее для каждого нового добавленного элемента реализуется новый класс. Для анимации элемента вызывается функция «**changeState**», аргументами которой является DOM-объект элемента, тип состояния, состояние и флаг (для отображения стилей). Определение типа элемента происходит по атрибуту «**xlink:href**».

2.6. Связь с другими программами

В процессе своей работы HTTP-шлюз подключается и поддерживает постоянную связь с сервером приложений ПК КОТМИ-14. Потеря связи приводит к останову и закрытию программы HTTP-шлюз.

2.7. Используемые методы

При разработке клиентского приложения в WEB-браузере, помимо стандартных возможностей JavaScript, была использована библиотека компонентов jQuery. Применены компоненты для работы с визуальным представлением табличных данных, паспортов свойств и иерархическими структурами. Это позволило унифицировать внешний вид и организацию работы с данными в программе.

3. НАСТРОЙКА ПРОГРАММЫ

3.1. Состав дистрибутива

В таблице 3.1 перечислен состав файлов дистрибутива с кратким описанием, которые будут размещены в папке установки после завершения процедуры инсталляции.

Таблица 3.1 - Состав файлов дистрибутива

<i>№</i>	<i>Файл</i>	<i>Описание</i>
1.	Серверная часть WEB-доступа	
1.1	.\bin\ScdWebServer.exe	WEB-сервер
1.2	.\bin\ScdWebServer.ini	Файл настроек HTTP-шлюза
1.3	.\bin\ScdMdx.dll	Библиотека для связи с сервером приложений ПК КОТМИ-14
1.4	.\bin\ScdWebServer.log	Log-файл
2.	Стандартные библиотеки Qt	
2.1	.\bin\Qt5Core.dll	
2.2	.\bin\Qt5Network.dll	
2.3	.\bin\Qt5Xml.dll	
3	Клиентская часть WEB-доступа	
3.1	.\src\web_files\auth.html	Начальная WEB-страница для авторизации
3.2	.\src\web_files\tabs.html	Основная WEB-страница
3.3	.\src\web_files\favicon.ico	Иконка для WEB-страницы
3.4	.\src\web_files\css	Папка для хранения css стилей
3.5	.\src\web_files\img	Папка для хранения кэша изображений
3.6	.\src\web_files\js	Папка для хранения js-скриптов
3.7	.\src\web_files\sht	Папка для хранения кэша документов
3.8	.\src\web_files\sounds	Папка для хранения кэша звуков
3.9	.\src\web_files\svg	Папка для хранения кэша схем

3.2. Установка программы

Для установки ПО «WEB-доступ» выполните следующие действия:

1. Запустите файл инсталлятора WEB-Server-2.0.XXXXXX.msi, где XXXXXX – номер редакции.
2. Выберите пункт **«Я принимаю условия лицензионного соглашения»** на странице с условиями лицензионного соглашения и нажмите кнопку **Далее**.
3. Выберите папку, в которую будет установлено ПО «WEB-доступ» и рабочую папку.
3. Нажмите кнопку **«Установить»**.
6. Нажмите кнопку **«Готово»** для окончания установки и завершения работы инсталлятора.

После завершения установки в меню запуска программ Windows и на рабочем столе создастся ярлык «Web-сервер КОТМИ-14».

3.3. Структура конфигурационного файла программы HTTP-шлюза

Программа HTTP-шлюза может использовать при запуске значения по умолчанию для всех параметров, необходимых для работы. Для задания произвольных значений параметров предусмотрен конфигурационный файл ScdWebServer.ini, находящийся в папке, указанной при установке программного пакета «WEB-доступ КОТМИ-14». Конфигурационный файл соответствует общепринятому формату текстового ini-файла в кодировке Windows-1251. Используемые в конфигурационном файле параметры приведены в таблице 3.2. При отсутствии параметра используется его значение по умолчанию.

Таблица 3.2 - Конфигурационные параметры АРМ

<i>Раздел</i>	<i>Параметр</i>	<i>Описание</i>	<i>Допустимые значения</i>	<i>Значение по умолчанию</i>
[scada]	address	Строка подключения к серверу приложений ПК КОТМИ-14 в виде IP-адрес[:порт1], где IP-адрес1[:порт1] — пара-	Строка	127.0.0.1:1312

		метры соединения с серверами приложений комплекса. Значением порта по умолчанию является 1312.		
[scada]	login	Имя пользователя для подключения к комплексу ПК КОТМИ-14	Строка	admin
[scada]	password	Пароль пользователя для подключения к комплексу ПК КОТМИ-14	Строка	
[params]	debug_level	Уровень вывода информации в лог-файл	0..5	4
[params]	lib_svg	Библиотека SVG-элементов для отображения схем	Строка	lib

В качестве символа комментария в ini-файле используется символ точки с запятой («;») в начале строки. Строки, отмеченные этим символом, игнорируются при обработке ini-файла.

3.4. Аргументы командной строки

Программу НТТР-шлюза можно запускать с использованием аргументов командной строки. Возможные параметры приведены в таблице (Таблица 3.3). При отсутствии параметра используется его значение по умолчанию.

Таблица 3.3 - Конфигурационные параметры программы НТТР-шлюза

<i>Параметр</i>	<i>Описание</i>	<i>Допустимые значения</i>	<i>Значение по умолчанию</i>
--help (-h, /help, /h)	Вывод на консоль справки для аргументов командной строки		
--address (-a, address=, a=)	Адрес сервера приложений (напр. 192.168.0.1:1312) ПК «КОТМИ-14»	Строка	127.0.0.1:1312
--login (-l, login=, l=)	Имя пользователя для подключения к комплексу ПК КОТМИ-14	Строка	admin

--password (-p, password=, p=)	Пароль пользователя для подключения к комплексу ПК КОТМИ-14	Строка	
--debug (-d, debug=, d=)	Уровень вывода информации в лог-файл	0..5	4
--svglib (-s, svglib=, s=)	Библиотека SVG-элементов для отображения схем	Строка	lib

3.5. Подключение через WEB-браузер к ПК КОТМИ-14

Для успешного выполнения функции подключения предварительно должен быть запущен сервер приложений ПК КОТМИ-14 и в файле ScdWebServer.ini HTTP-шлюза указан IP-адрес этого сервера.

После запуска, программа HTTP-шлюза осуществит поиск и считывание информации из ini-файла в «рабочем» каталоге. С учетом полученной информации будет выполнено подключение к серверу приложений ПК КОТМИ-14.

В браузере вводится строка подключения с IP-адресом и портом HTTP-шлюза. После этого в браузере выводится окно (Рисунок 3.1).

The image shows a web browser window displaying a login form. The form has two input fields: 'Логин' (Login) with the text 'user' and 'Пароль' (Password) with the text 'password'. Below the fields is a blue button labeled 'Отправить' (Send).

Рисунок 3.1 - Форма входа в систему

После задания логина и пароля, нажатия кнопки «Отправить» откроется основное окно программы. В адресной строке будет указан сервер, к которому подключен WEB-браузер (Рисунок 3.2).



Рисунок 3.2 - Основное окно WEB-доступа после входа в систему

В случае невозможности подключения, в зависимости от причины, программа HTTP-шлюза или WEB-браузер высветит диагностическое сообщение и ПО «WEB-доступ» завершит свою работу.

A screenshot of a login form. It has two input fields: "Логин" (Login) with the value "admin1" and "Пароль" (Password) with the value "password". Below the fields, there is a red error message "Логин неверен" (Login is incorrect). At the bottom, there is a blue button labeled "Отправить" (Send).

Рисунок 3.3 - Извещение о невозможности подключения

4. НАСТРОЙКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Клиентская часть ПО «WEB-доступ» предоставляет пользователю возможность работать в WEB-браузере с интерфейсом, поддерживающий одновременное открытие множества вкладок. Главное окно программы состоит из следующих частей: основного меню и области отображения вкладок окна.

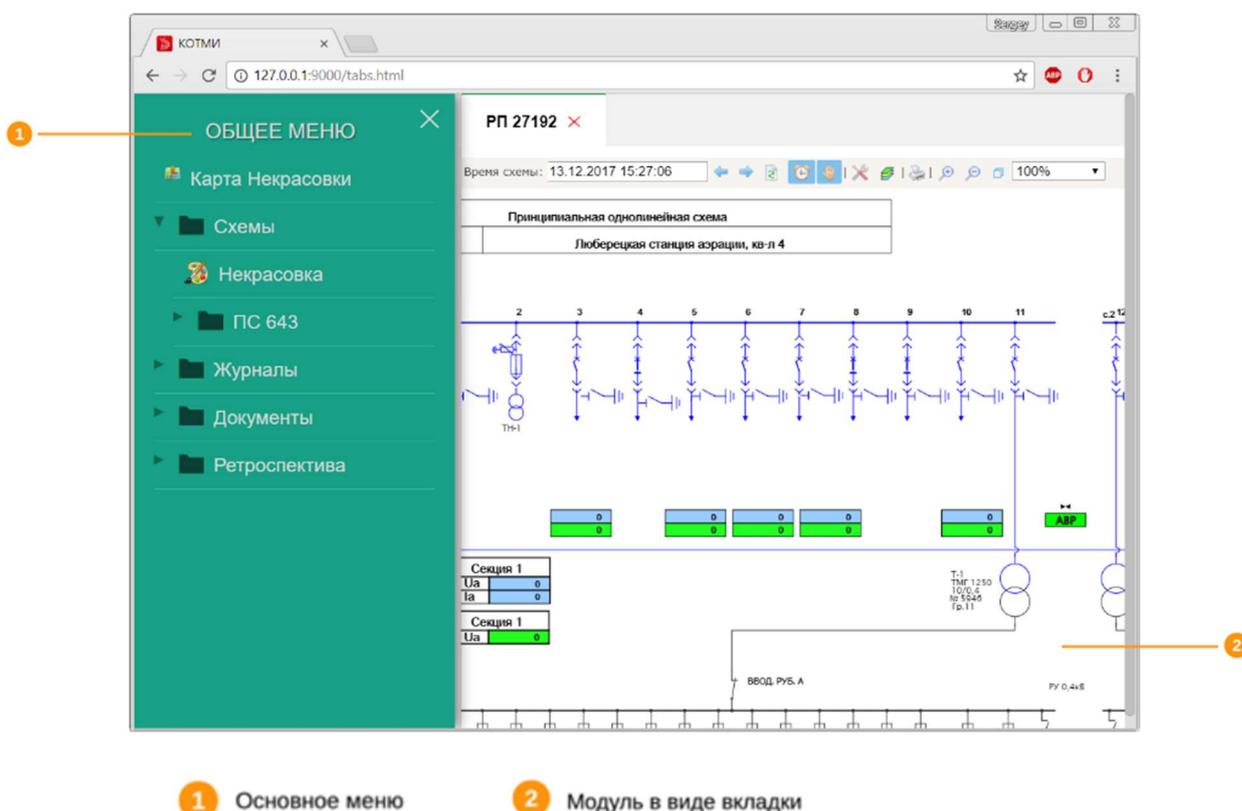


Рисунок 4.1 - Оконный интерфейс программы

Состав действий, доступных на панели задач, настраивается и определяется правами пользователя, авторизованного в программе.

В процессе работы пользователь взаимодействует с различными функциональными формами общего назначения. К формам общего назначения можно отнести схемы, таблицы, отчеты. Перед использованием все эти документы должны быть подготовлены и настроены в соответствующих редакторах ПК КОТМИ-14.

5. ДОБАВЛЕНИЕ ЭЛЕМЕНТОВ В БИБЛИОТЕКУ SVG

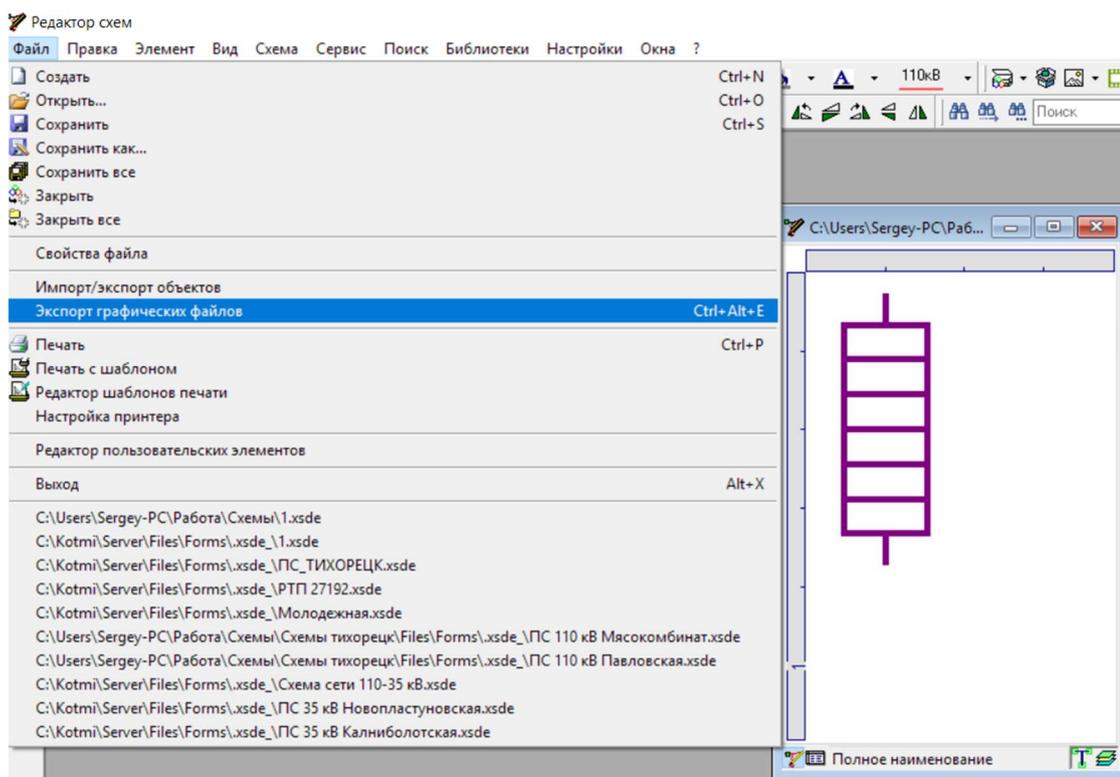
Когда тип элемента при конвертации схем из формата компании «Модус» не определён, то в лог-файл пишется строчка следующего содержания:

```
18.03.15 13:00:29.365 (1) CDomXSDEParser:parseType() Unknown type of element:  
(RAZRYADNIK_500_CUS) 320
```

Где два последних аргумента это название объекта и номер типа.

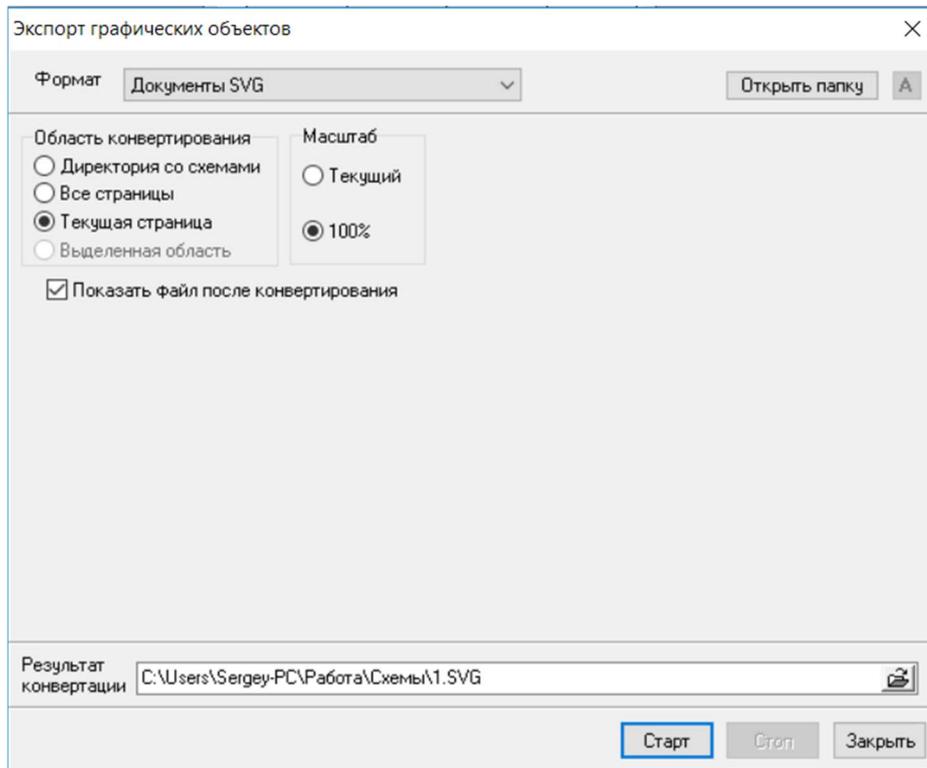
Для конвертации объекта необходимо:

- С помощью «Графического редактора схем» компании «Модус» переместить необходимый объект в новый файл.
- С помощью любого текстового редактора удалить атрибуты «**orient**» и «**angle**» и сохранить файл.
- С помощью «Графического редактора схем» компании «Модус» закрыть и открыть изменённый файл.
- Выбрать пункт меню «**Файл -> Экспорт графических файлов**».

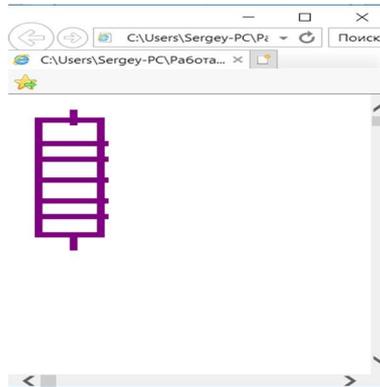


- В появившемся окне выбрать следующие настройки:
 - «Формат -> Документ SVG».
 - «Область конвертирования -> Текущая страница».

- «Масштаб -> 100%».
- «В Результат конвертации» указать путь к выходному SVG-файлу.



После успешного конвертирования элемента в формат SVG откроется браузер с данным элементом.



С помощью любого текстового редактора необходимо открыть данный файл.

Файл будет примерно следующего содержания:

```

1 <?xml version="1.0" encoding="windows-1251"?>
2 <svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" width="793.6992" height="1122.5184">
3 <title> </title>
4 <desc> </desc>
5 <rect x="4.0000" y="5.0000" width="8.0000" height="22.0000" fill="none" style="stroke:#7F0080; stroke-width:1;" />
6 <rect x="4.0000" y="9.0000" width="9.0000" height="1.0000" fill="#7F0080" />
7 <rect x="4.0000" y="20.0000" width="9.0000" height="1.0000" fill="#7F0080" />
8 <rect x="8.0000" y="27.0000" width="1.0000" height="3.0000" fill="#7F0080" />
9 <rect x="4.0000" y="12.0000" width="9.0000" height="1.0000" fill="#7F0080" />
10 <rect x="4.0000" y="16.0000" width="9.0000" height="1.0000" fill="#7F0080" />
11 <rect x="4.0000" y="23.0000" width="9.0000" height="1.0000" fill="#7F0080" />
12 <rect x="8.0000" y="3.0000" width="1.0000" height="3.0000" fill="#7F0080" />
13 </svg>
14

```

Для уменьшения размера файла SVG-библиотеки необходимо убрать в числовых значениях все знаки, которые не влияют на значения. В данном случае «.0000»

Далее необходимо убрать все атрибуты «**fill**», «**stroke**» и «**style**» у всех элементов, кроме фигур, которые не должны иметь заливки. В данном случае это первый прямоугольник.

Для визуального представления можно добавить все элементы в группу с атрибутами «**fill**» и «**stroke**».

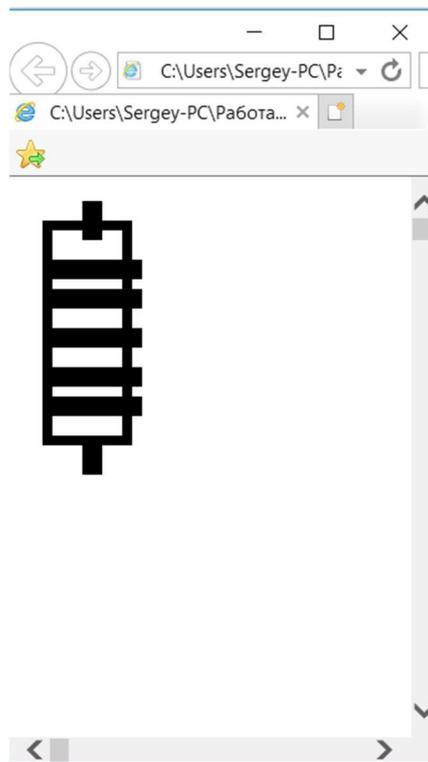
Файл будет выглядеть следующим образом:

```

5 <g fill="black" stroke="black">
6   <rect x="4" y="5" width="8" height="22" fill="none"/>
7   <rect x="4" y="9" width="9" height="1"/>
8   <rect x="4" y="20" width="9" height="1"/>
9   <rect x="8" y="27" width="1" height="3"/>
10  <rect x="4" y="12" width="9" height="1"/>
11  <rect x="4" y="16" width="9" height="1"/>
12  <rect x="4" y="23" width="9" height="1"/>
13  <rect x="8" y="3" width="1" height="3"/>
14 </g>

```

А отображение самого элемента следующим:

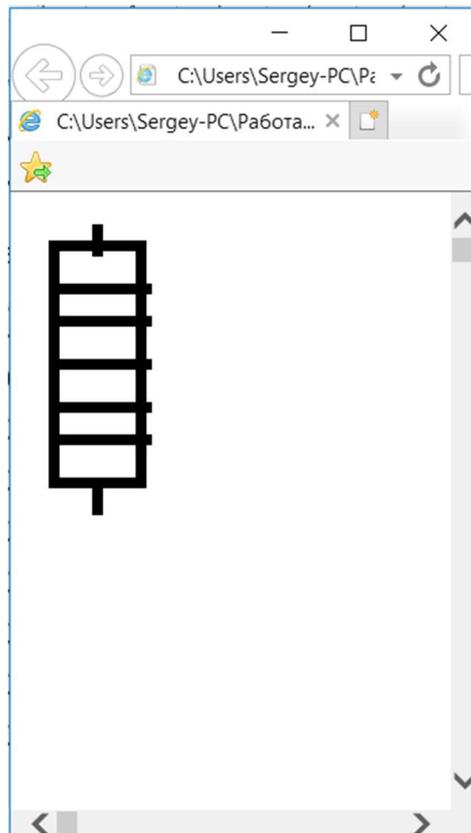


Далее необходимо заменить все прямоугольники, у которых высота или ширина равна 1, соответствующими линиями.

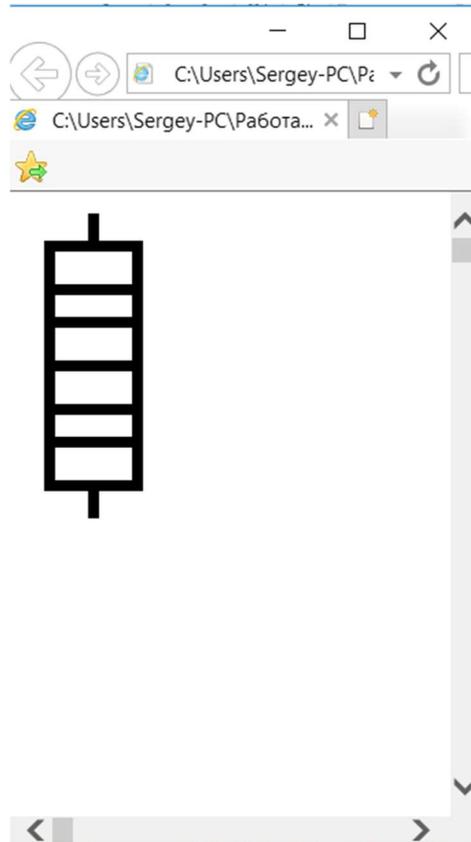
Файл приобретёт вид:

```
5 <g fill="black" stroke="black">
6   <rect x="4" y="5" width="8" height="22" fill="none"/>
7   <line x1="4" y1="9" x2="13" y2="9"/>
8   <line x1="4" y1="20" x2="13" y2="20"/>
9   <line x1="8" y1="27" x2="8" y2="30"/>
10  <line x1="4" y1="12" x2="13" y2="12"/>
11  <line x1="4" y1="16" x2="13" y2="16"/>
12  <line x1="4" y1="23" x2="13" y2="23"/>
13  <line x1="8" y1="3" x2="8" y2="6"/>
14 </g>
```

А сам элемент:



Далее необходимо подкорректировать значения, для получения лучшей картинки.



Файл будет выглядеть следующим образом:

```

5 <g fill="black" stroke="black">
6   <rect x="4" y="5" width="8" height="22" fill="none"/>
7   <line x1="4" y1="9" x2="12" y2="9"/>
8   <line x1="4" y1="20" x2="12" y2="20"/>
9   <line x1="8" y1="27" x2="8" y2="30"/>
10  <line x1="4" y1="12" x2="12" y2="12"/>
11  <line x1="4" y1="16" x2="12" y2="16"/>
12  <line x1="4" y1="23" x2="12" y2="23"/>
13  <line x1="8" y1="2" x2="8" y2="5"/>
14 </g>

```

Далее необходимо элемент переместить в крайний левый угол с отступом в 2 единицы.

Файл после:

```

5 <g fill="black" stroke="black">
6   <rect x="2" y="5" width="8" height="22" fill="none"/>
7   <line x1="2" y1="9" x2="10" y2="9"/>
8   <line x1="2" y1="20" x2="10" y2="20"/>
9   <line x1="6" y1="27" x2="6" y2="30"/>
10  <line x1="2" y1="12" x2="10" y2="12"/>
11  <line x1="2" y1="16" x2="10" y2="16"/>
12  <line x1="2" y1="23" x2="10" y2="23"/>
13  <line x1="6" y1="2" x2="6" y2="5"/>
14 </g>

```

И добавить описанный прямоугольник так, чтобы элемент находился по центру данного прямоугольника.

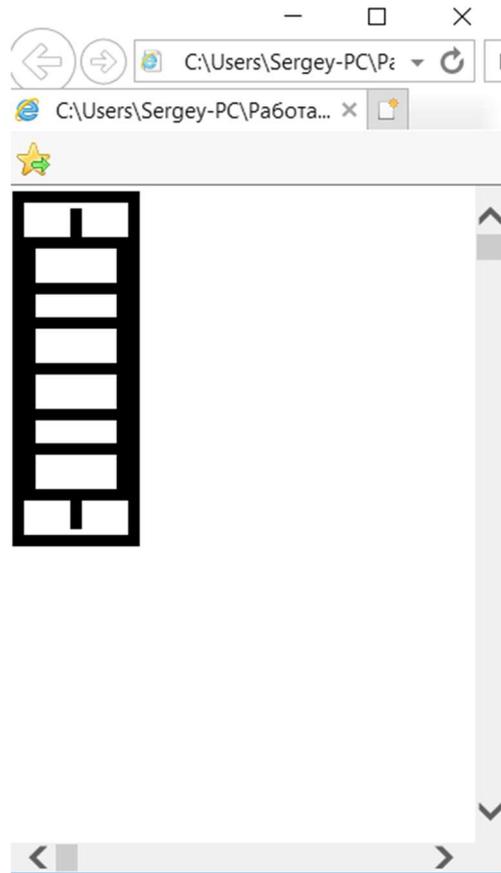
Файл:

```

5 <g fill="black" stroke="black">
6   <rect x="1" y="1" width="10" height="30" fill="none"/>
7   <rect x="2" y="5" width="8" height="22" fill="none"/>
8   <line x1="2" y1="9" x2="10" y2="9"/>
9   <line x1="2" y1="20" x2="10" y2="20"/>
10  <line x1="6" y1="27" x2="6" y2="30"/>
11  <line x1="2" y1="12" x2="10" y2="12"/>
12  <line x1="2" y1="16" x2="10" y2="16"/>
13  <line x1="2" y1="23" x2="10" y2="23"/>
14  <line x1="6" y1="2" x2="6" y2="5"/>
15 </g>

```

Изображение:



Далее необходимо добавить элемент в файл SVG-библиотеки. Для этого необходимо создать группу с атрибутом «**type**», в котором указывается номер типа, название класса. В данной группе создать тег «**symbol**» с атрибутом «**id="НАЗВАНИЕ_КЛАССА_simple"**». В данный символ вставить все примитивы. Заменить в описывающем прямоугольнике атрибут «**fill="none"**» атрибутом «**class="substrate invisible"**».

```

<!--Разрядник-->
<g class="discharger_320" type="320">
  <symbol id="discharger_320_simple" preserveAspectRatio="xMidYMid meet">
    <rect x="1" y="1" width="10" height="30" class="substrate invisible"/>
    <rect x="2" y="5" width="8" height="22" fill="none"/>
    <line x1="2" y1="9" x2="10" y2="9"/>
    <line x1="2" y1="20" x2="10" y2="20"/>
    <line x1="6" y1="27" x2="6" y2="30"/>
    <line x1="2" y1="12" x2="10" y2="12"/>
    <line x1="2" y1="16" x2="10" y2="16"/>
    <line x1="2" y1="23" x2="10" y2="23"/>
    <line x1="6" y1="2" x2="6" y2="5"/>
  </symbol>
</g>

```

Далее в файле xsdeObjects.h проекта web обозначить тип

```
#define XSDE_OBJ_PRESENCE_EM_DRIVE
```

```
320
```

В функции `parseType` класса `CDomXSDEParser` в разделе определения типа добавить номер типа в нужный `case`. В данном случае в тот `case`, где создаётся объект класса `SimpleObject`.

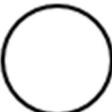
6. ПОДГОТОВКА СХЕМ ДЛЯ ОТОБРАЖЕНИЯ В WEB-БРАУЗЕРЕ

Модуль «Схемы» предназначен для отображения интерактивных схем объектов автоматизации ПК КОТМИ-14. Графические элементы схем являются динамическими, то есть меняют свою форму и цвет в зависимости от связанных с ними значений сигналов.

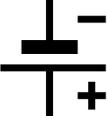
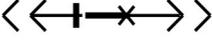
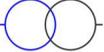
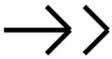
В модуле «Схемы» используется формат SVG, полученный автоматической конвертацией из формата схем программного комплекса «Модус». Программа HTTP-шлюза сравнивает файлы схем на сервере ПК КОТМИ-14 и в кэше HTTP-шлюза по времени последней модификации. При различии данного времени происходит автоматическая конвертация схемы. При совпадении схема берётся из кэша HTTP-шлюза. Папка, хранящая кэш SVG-схем, находится по адресу.\src\web_files\svg.

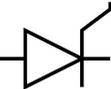
Для конвертации схем необходима библиотека элементов в формате SVG. Основные элементы представлены в таблице (Таблица 6.1).

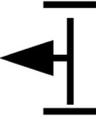
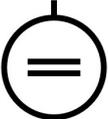
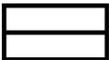
Таблица 6.1 - Перечень поддерживаемых элементов «МОДУС»

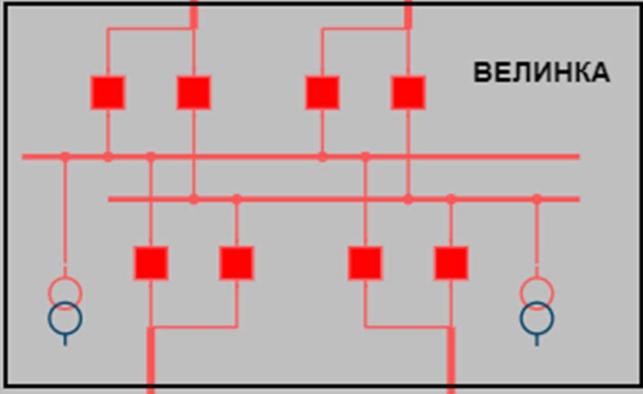
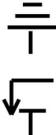
<i>Название элемента</i>	<i>Пример отображения</i>
линия	
стрелка	
прямоугольник	
круг	
текст	ПС 220 кВ Койсуг
вдт (трансформатор с одной обмоткой)	
точка	

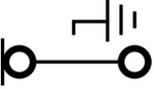
<i>Название элемента</i>	<i>Пример отображения</i>
связь	
коннектор	
непересечение	
многоугольник	
крепление	
ошиновка	
воздушная_линия	
кабельная_линия	
шина	
связь_с_объектом	
ОПН	
заземление	
муфта	

<i>Название элемента</i>	<i>Пример отображения</i>
дроссельная_катушка	
трансформатор_тока	
разрядник	
реактор	
подстанция_	
двигатель асинхронный	
батарея	
выключатель	
выключатель_нагрузки	
выключатель_выдвижной	
разъединитель_выдвижной, от- делитель_выдвижной	
автотрансформатор или трансфор- матор_силовой	
шасси с 3-ёх фазным включением	
полушасси	

<i>Название элемента</i>	<i>Пример отображения</i>
заземляющий_нож	
трансформатор_напряжения	
кабельный_разъем	
масляный выключатель	
присоединение_Р2	
присоединение_РЗД	
ТП_комп_не_сек	
присоединение_с_яч_КРУ	
символ_АВР_2	
символ_УТКЗ	
автоматика, табло	
лампа	
устройство	
прибор_цифровой2	
предохранитель_на_тележке	
тиристор	

<i>Название элемента</i>	<i>Пример отображения</i>
рубильник	
разъединитель	
короткозамыкатель_без_земли	
отделитель	
ОПН1	
генератор БСК	
генератор	
синхронный_компенсатор	
рубильник_3-позиц	
предохранитель	
привод	

<i>Название элемента</i>	<i>Пример отображения</i>																																			
контейнер																																				
таблица, таблица2	<table border="1" data-bbox="735 707 1444 958"> <thead> <tr> <th colspan="5">Схема ПС 110/10кВ "Ботаническая"</th> </tr> <tr> <th></th> <th>Должность</th> <th>Ф.И.О.</th> <th>Подпись</th> <th>Дата</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Составил</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Чертил</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Схема ПС 110/10кВ "Ботаническая"						Должность	Ф.И.О.	Подпись	Дата																Составил					Чертил				
Схема ПС 110/10кВ "Ботаническая"																																				
	Должность	Ф.И.О.	Подпись	Дата																																
Составил																																				
Чертил																																				
(Наличие_ЭМ_привода)																																				
КТП																																				
ЗТП																																				
конденсатор																																				
заградительный_фильтр																																				
реактор_шунтирующий																																				
короткозамыкатель																																				

<i>Название элемента</i>	<i>Пример отображения</i>
автомат_силовой	
3х_позиционный_КА	

7. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входными данными для программы HTTP-шлюза является ini-файл программы, считываемый на этапе загрузки. Из файла ScdWebServer.ini HTTP-шлюз берет информацию о сервере приложений ПК КОТМИ-14, к которому необходимо осуществить подключение, имя и пароль пользователя, и другие настройки.

```
[scada]
address=127.0.0.1:1312
login=admin
password=
```

```
[params]
debug_level=4
lib_svg=libTihorezk
```

В процессе работы программа HTTP-шлюза подключается к серверу приложений и, в зависимости от информационных запросов пользователей, считывает требуемую информацию из базы данных ПК КОТМИ-14.

Входными данными для программы HTTP-шлюза является также библиотека элементов пользователя, необходимая для конвертации схем МОДУС в формат SVG.

Выходными данными для ПО «WEB-доступ» являются формы отображения в формате WEB-страниц.

8. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

В процессе работы программы HTTP-шлюза, при возникновении ошибочных ситуаций, программа фиксирует данные события путем записи соответствующих диагностических сообщений в файл ScdWebServer.log.

Формат сообщений единообразен и представляет собой отдельную строку текста с указанием времени добавления сообщения. Например,

10/11/12 17:27:21=> Нет доступа к таблице "T_DISPLAY_LIB".

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

АРМ – автоматизированное рабочее место;

БД - база данных

НСИ – нормативно справочная информация

ОС - операционная система

ПК - программный комплекс

ПО - программное обеспечение

ЯКШГ.00067-01 91 01-05 32

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					

Инв.№ подл.	Подп. и дата	Взам. инв. №	Инв.№ дубл.	Подп. и дата