

УТВЕРЖДЕНО

СТ.2019.SCADA.ТД.00002-ЛУ

## SCADA-ПАКЕТ SCAD CC

### Описание программы

СТ.2019.SCADA.ТД.00002

Листов 77

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

## СОДЕРЖАНИЕ

Введение .....	4
1 Общие сведения о программе .....	5
1.1 Назначение и функции программы .....	5
1.2 Термины и сокращения .....	5
1.3 Архитектура программы .....	9
1.4 Событийно-ориентированный подход.....	12
1.5 Резервирование, распределенные системы .....	12
1.6 Пользовательские интерфейсы .....	13
1.7 Концепция точек данных, образ процесса .....	17
1.8 Общие сведения о драйверах .....	19
2 Модель данных.....	21
2.1 Точки данных как носители информации .....	21
2.2 Адресация .....	25
2.3 Конфигурационные элементы и мастер-точки данных .....	25
3 Создание и настройка точек данных .....	27
3.1 Создание типов точек данных .....	27
3.2 Создание точек данных .....	28
3.3 Создание мастер-точки данных .....	30
3.4 Настройка элементов точек данных.....	31
3.5 Импорт настроек с использованием ASCII-файлов (массовое параметрирование).....	32
3.6 Создание списков точек данных .....	35
3.7 Создание типов точек данных и точек данных с использованием сценариев .....	36
4 Создание экранных форм .....	37
4.1 Графический редактор.....	37
4.2 Меню пиктограмм.....	37
4.3 Основные операции в редакторе .....	43
4.4 Редактор свойств.....	44
4.5 Модули, панели, дочерние панели .....	45
5 Алармы.....	47
5.1 Настройка обработки алармов.....	47
5.2 Состояния аларм-диапазона и квитирование.....	49
5.3 Цвета алармов, мигание .....	50
5.4 Панель алармов .....	51
6 Архивирование .....	54
6.1 Активация ведения истории значений.....	54
6.2 Отображение исторических данных с помощью трендов .....	56
7 Создание сценариев .....	57
7.1 Общая информация.....	57
7.2 Редактор сценариев Control .....	58

7.3 Структура сценариев и синтаксис .....	59
7.4 Переменные, константы, типы данных.....	60
7.5 Управляющие структуры .....	61
7.6 Программы, функции, библиотеки .....	64
7.7 Адресация элементов точек данных, конфигурационных элементов, атрибутов .....	67
7.8 Операции со значениями элементов точек данных.....	69
7.9 Операции со значениями свойств графических объектов .....	73
7.10 Менеджеры сценариев.....	75
Перечень сокращений.....	77

## **ВВЕДЕНИЕ**

Настоящий документ содержит справочные сведения о программе, ее архитектуре, а также описывает основы использования программы.

## 1 Общие сведения о программе

### 1.1 Назначение и функции программы

Программный SCADA-пакет SCAD CC предназначен для использования в области автоматизации технологического процесса. Основная сфера применения – работа с программным обеспечением и эксплуатация технологического оборудования посредством АРМ.

Благодаря модульной концепции SCADA-пакет представляет собой многофункциональную и перспективную платформу для создания систем управления любого типа. За счет последовательного использования архитектуры «клиент-сервер» в системе поддерживается многопользовательский доступ. Поддерживаются различные варианты установки системы: от установки на одном компьютере с минимальными требованиями к системе, до установки на нескольких высокопроизводительных компьютерах с целью распределения нагрузки (так называемые «распределенные системы»). Горячее резервирование серверов в сочетании с полным резервированием сети обеспечивают высокий уровень отказоустойчивости.

Основные функции программы:

- графическое представление текущего состояния технологического процесса;
- возможность передачи состояния и команды оборудованию по контролю и управлению технологическим процессом;
- оповещение пользователя в случае превышения предельных значений и наступления аварийного состояния;
- архивирование исторических данных для отображения и использования их в будущем.

Совместно с системами базовой автоматизации, а также соответствующими датчиками и модулями ввода и вывода программа образует полнофункциональную систему автоматизации.

### 1.2 Термины и сокращения

Сокращение	Полная форма	Значение
ASCII	Менеджер ASCII	Менеджер, позволяющий осуществлять импорт и экспорт структур и настроек точек данных и типов точек данных по стандартизованному протоколу хранения и передачи символов и текста. Для обмена данными с менеджером ASCII используются ASCII-файлы (CSV), в связи с чем исходная информация для проведения массового параметрирования может быть подготовлена в программе для работы с электронными таблицами (например, MS Excel) или во внешней базе данных
API	Application Programming Interface – Интерфейс прикладного программирования	Интерфейс API позволяет интегрировать в проект собственные решения и алгоритмы в виде новых менеджеров

Сокращение	Полная форма	Значение
Config	Конфигурационный элемент	Для каждого элемента точки данных могут настраиваться конфигурационные элементы, определяющие набор свойств элемента точки данных и связанных с ним функций. К таковым могут относиться, например, проверка допустимого диапазона значений, параметры обработки алармов, настройки архивирования
CTRL	Control Manager – Менеджер сценариев	Интерпретатор, позволяющий выполнять пользовательские программы (сценарии). Язык Control отличается простым синтаксисом (схож с ANSI-C). Написанные на нем сценарии выполняются интерпретатором. Менеджер поддерживает многопоточность, множество различных интерфейсов (SMTP, COM, HTTP, DDE...), а также обеспечивает доступ к обширной библиотеке функций SCADA. Набор функций может расширяться пользователем путем подключения сторонних библиотек, разработанных на языке Control, или путем подключения скомпилированных динамически подключаемых библиотек, разработанных на C++ (интеграция DLL)
D	Driver – Менеджер драйвера	Интерфейс для обеспечения связи с контроллерами (ПЛК, ПЦУ, ...), полевыми шинами и устройствами телеуправления. С помощью драйвера данные, передаваемые в соответствии с внешним протоколом, переводятся в формат, применяемый в платформе SCAD CC. При этом применяется событийно-ориентированный подход, возможно сравнение старых и новых значений, сглаживание, конвертирование, конвертирование типов и т.д. Драйвер должен соответствовать управляющему устройству или шине связи. Данное требование необходимо учитывать при выборе драйвера. Помимо прочего, в SCAD CC доступны следующие драйверы: Modbus TCP, SSI, МЭК 60870-5-101 и -104, OPC DA, OPC A&E, SNMP и т.д. Идентификация передаваемого значения осуществляется в том числе с помощью адреса периферийного устройства, привязанного к соответствующему элементу точки данных
DIST	Distribution Manager – Менеджер распределенной архитектуры	Модуль для обеспечения коммуникации с одной или несколькими системами SCAD CC. Обеспечивает доступ к текущим данным, алармам и историческим данным других систем без дублирования точек данных в собственной системе и позволяет избежать дополнительных затрат на разработку средств для обмена данными

Сокращение	Полная форма	Значение
DM	Data Manager – Менеджер базы данных, Менеджер БД, Менеджер данных (взаимодействие с базами данных)	Менеджер базы данных обеспечивает наличие интерфейсов к базам данных: 1) обеспечивает архивирование истории значений. В качестве хранилища используются либо «архив значений» (value archive, VA), либо РБД 2) обеспечивает архивирование истории состояний алармов (в БД алармов) 3) обеспечивает инициализацию всех других менеджеров при запуске системы (в основе лежит БД конфигураций). Данная БД содержит информацию о структуре модели данных, т.е. обо всех точках данных, обработке алармов, диапазонах значений и т.д.
DP	Data point – Точка данных	Структурированный и связанный с конкретным устройством виртуальный объект (носитель информации), который служит для представления реального устройства в системе управления. Точка данных содержит один или несколько элементов точки данных (переменных процесса)
DPA	Data point attribute – Атрибут точки данных	Помимо текущего значения переменной технологического процесса, каждый элемент точки данных хранит следующие атрибуты – как минимум, информацию о качестве сигнала, метку времени и адрес
DPE	Data point element – Элемент точки данных	Отдельная единица информации о процессе в структуре точки данных, связанной с конкретным устройством. Каждый DPE отражает какое-либо значение или состояние. Помимо значения, DPE хранит такие атрибуты, как метка времени, информация о качестве сигнала, адрес
DPT	Data point type – Тип точки данных	Определение объекта (класс), задающее структуру для точек данных, которые соответствуют реальным устройствам конкретного типа. Отдельные точки данных (экземпляры) наследуются от соответствующего Типа точки данных. Тип точек данных является своего рода шаблоном структуры для точек данных
EV	Event Manager – Менеджер событий («диспетчер данных»)	Менеджер событий является центральным узлом системы. Менеджер содержит и поддерживает в актуальном состоянии образ процесса (т.е. набор переменных и их значения). Образ процесса иногда называют «Run Time Database». Менеджер событий обновляет образ процесса на основании информации от драйверов и, в свою очередь, передает драйверам команды и значения уставок. Менеджер событий также обеспечивает обработку алармов и авторизацию пользователей на уровне точек данных
GEDI	Graphic Editor GEDI – Графический редактор	Графический редактор. Используется для разработки экранных форм («панелей»), а также для создания диалоговых окон и примитивов
HMI	Human Machine Interface – Человеко-машинный интерфейс, ЧМИ	Аппаратное и программное обеспечение, применяемое в компьютерных решениях для мониторинга и управления процессами

Сокращение	Полная форма	Значение
I/O	Input/Output – Ввод/вывод (вход/выход)	В рамках систем SCADA под данным термином понимается информация, передаваемая или принимаемая от внешних устройств, таких как ПЛК
LV	Log Viewer – Средство просмотра журналов (системный журнал)	Средство для просмотра выходных системных сообщений менеджеров
MV-P	Модуль VISION – PARA (редактор БД)	Особый вид модуля VISION пользовательского интерфейса, используемый в качестве редактора БД. Данный модуль, как правило, обозначается сокращением PARA. С его помощью создаются и настраиваются связанные с устройствами точки данных и определяются специальные функции. Модуль PARA также выполняет роль обозревателя базы данных
OLE-DB	OLE-DB Provider – Поставщик данных OLE-DB (Интерфейс БД)  OLE-DB – Object Linking and Embedding-Database («связывание и встраивание объектов, база данных» – технология Microsoft для доступа к базам данных)	Стандартизованный интерфейс баз данных архивов значений (VA) и алармов. С помощью поставщика данных OLE-DB внешним программам обеспечивается доступ к данным процесса и алармам SCAD CC, как если бы они были сохранены в стандартной реляционной БД
PARA	Инструмент для конфигурирования	Редактор БД для создания типов точек данных, собственно точек данных, а также для их конфигурирования
PLC	Programmable logic controller – Программируемый логический контроллер, ПЛК	Промышленная компьютерная система для управления процессами
PMON	Process Monitor – Монитор процессов (мониторинг системы)	Фоновый процесс, используемый для мониторинга и управления запуском и работой менеджеров
REDU	Redundancy Manager – Менеджер резервирования (подключение резервирующей системы)	Менеджер резервирования предназначен для подключения идентичной системы SCAD CC, используемой в качестве горячего резерва
RTU	Remote Terminal Unit, Remote Telemetry Unit – Устройство связи с объектом, УСО	Удаленная система управления
SCADA	Supervisory Control And Data Acquisition – Система диспетчерского управления и сбора данных	Ведущая система для управления технологическим процессом и сбора данных. Также обозначает технологию управления и визуализации
UI	User Interface – Интерфейс пользователя (визуализация)	Графический интерфейс пользователя для мониторинга и управления технологическим процессом. Как правило, с помощью данного программного модуля обеспечивается взаимодействие пользователя и процесса. Кроме того, модуль VISION обеспечивает отображение трендов (графиков изменений значений во времени), а также поддерживает панель алармов, отчетность и, при необходимости, конфигурирование в режиме выполнения проекта



Сокращение	Полная форма	Значение
VA	Value Archive – Архив значений (БД значений)	Архив истории значений. Менеджером базы данных одновременно могут обслуживаться несколько таких архивов
VISION	Модуль визуализации	Сокращенное название пользовательского интерфейса «Модуль VISION»
SCAD CC	-	Название программного продукта от компании АО «СКАД тех» для решения задач визуализации и управления

### 1.3 Архитектура программы

SCAD CC представляет собой модульную систему: отдельные функции системы реализуются на базе отдельных программных модулей, которые в системе SCAD CC именуются «менеджерами». Менеджеры являются отдельно выполняемыми программными процессами. Упрощенная конфигурация приведена на рисунке ниже.

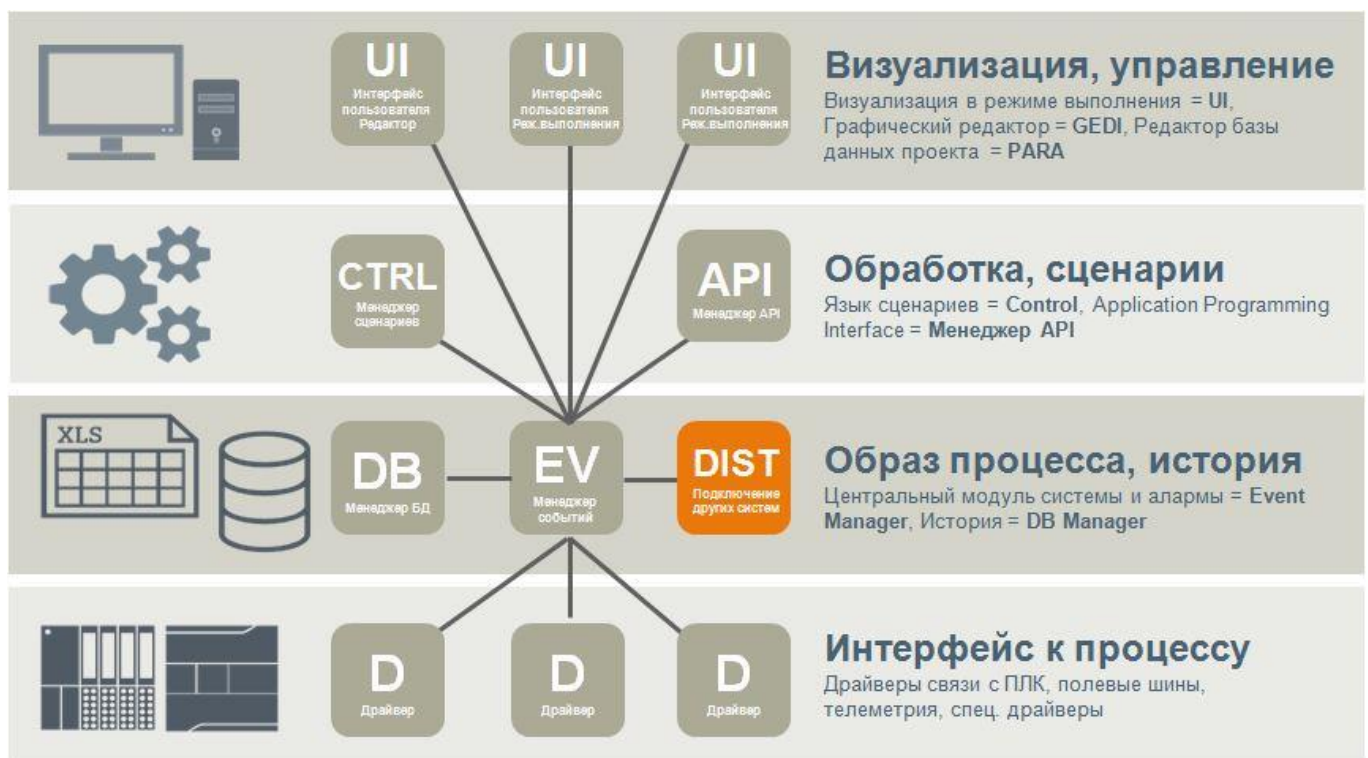


Рисунок 1 – Модульная архитектура системы SCAD CC

Модули интерфейса взаимодействия с процессом, которые именуются драйверами (D), образуют нижний уровень системы SCAD CC. Драйверы представляют собой специализированные программы, обеспечивающие взаимодействие системы SCAD CC с полевыми устройствами. В связи с существованием множества различных протоколов для обеспечения связи с ПЛК и удаленными узлами управления, в состав системы SCAD CC включен набор различных драйверов.

Выбор драйвера в каждом конкретном случае определяется ПЛК и/или шиной, с которыми необходимо организовать связь. Драйвер представляет собой программный модуль, с помощью которого данные, передаваемые по какому-либо протоколу, преобразуются во внутренний формат данных SCAD CC. При помощи драйверов с полевых устройств считываются текущие состояния, измеренные значения или значения счетчиков. Также при помощи драйверов на нижестоящие устройства управления передаются команды и уставки.

Центральным модулем системы является менеджер событий (Event Manager, EV). Менеджер содержит и поддерживает в актуальном состоянии образ процесса (т.е. набор переменных и их значения). Когда какой-либо другой функциональный модуль (менеджер) запрашивает какие-либо данные, эти данные передаются ему из образа процесса в менеджере событий (без прямого взаимодействия с устройствами управления). При поступлении команды с пульта оператора, значение соответствующей переменной сначала изменяется в образе процесса в менеджере событий. Далее перенаправление команды на целевое устройство (например, ПЛК) выполняется соответствующим менеджером автоматически.

Менеджер событий исполняет роль центрального диспетчера данных и центрального узла взаимодействия в системе SCAD CC. Менеджер событий также обеспечивает обработку алармов и может самостоятельно выполнять различные вычисления.

Менеджер событий работает в связке с менеджером базы данных. Менеджер БД обеспечивает взаимодействие с базой данных. С его помощью обеспечивается доступ к хранящимся в БД настройкам приложения, а также обеспечивается управление историей значений и историей алармов, которые также хранятся в БД. Пользовательские запросы на получение исторических данных также обрабатываются менеджером данных, а не самой базой данных.

Концепция архивирования данных процесса подразумевает сохранение и последующее чтение информации, поступающей или формируемой в ходе управления и визуализации. При этом могут архивироваться изменения значений, а также сообщения, формируемые при этих изменениях. Данные процесса сохраняются в архивах значений (VA, Value Archives). Каждый архив управляется с помощью отдельного процесса. Каждый архив состоит из серии расположенных в хронологическом порядке архивных файлов.

SCAD CC предоставляет множество возможностей для реализации собственных алгоритмов и процедур обработки данных. Наиболее важными компонентами при этом являются встроенный язык программирования Control (CTRL), а также API (интерфейс прикладного программирования).

Control представляет собой язык сценариев. Созданный на нем код обрабатывается интерпретатором, в связи с чем компиляция (процесс предварительного перевода исходного кода в машинный код для последующего исполнения процессором) не требуется. Язык отличается

простым синтаксисом (схож с ANSI-C) и представляет собой язык программирования высокого уровня с поддержкой многопоточности (параллельного выполнения отдельных программ под управлением системы). Язык обладает обширной библиотекой функций для задач управления и визуализации. Сценарии на языке Control могут выполняться в рамках менеджера сценариев (в качестве отдельного процесса), в рамках менеджера интерфейса пользователя (для анимации и в процессе разработки), а также в рамках менеджера событий.

API обеспечивает возможности для расширения функциональности системы. API выполнен в виде библиотеки классов C++ и позволяет разработчику ПО включать в систему собственную функциональность (например, системы прогнозирования, симуляции, планирования, взаимодействия с проприетарными БД и т.д.) в виде дополнительных менеджеров.

Менеджер пользовательского интерфейса (UI) обеспечивает возможность взаимодействия пользователя с системой. Формами интерфейса пользователя являются графический редактор (GEDI), редактор базы данных (PARA), а также интерфейс отображения экранных форм (модуль VISION). С помощью пользовательского интерфейса выводятся значения, отображаются и квитируются алармы, а также формируются команды. Тренды и отчеты обычно также являются частью пользовательского интерфейса. С технической точки зрения, выводимая пользователю информация в SCAD CC полностью отделена от обработки данных, выполняемой в фоновом режиме, и представляет собой «взгляд пользователя» на актуальный образ процесса или исторические данные.

Взаимодействие между отдельными менеджерами осуществляется на основе клиент-серверной архитектуры, что предполагает независимую от клиента обработку и предоставление данных серверами. Серверы в данной концепции являются поставщиками информации. Связь между менеджерами осуществляется по протоколу TCP/IP.

Клиенты, соответственно, являются получателями (потребителями) информации. Клиенты получают информацию от серверов. Подобное взаимодействие иногда называют взаимодействием между поставщиками и потребителями (Provider-Consumer).

Структура, состоящая из менеджера событий SCAD CC, менеджера данных SCAD CC, а также нескольких других менеджеров, называется системой SCAD CC. Менеджер событий и менеджер базы данных образуют отдельную законченную работоспособную систему, которая может быть дополнена, например, драйвером (D) для взаимодействия с внешними устройствами.

Все остальные менеджеры, такие как, например, интерфейс пользователя (UI) или менеджер сценариев (CTRL), могут добавляться по необходимости. Таким образом обеспечивается масштабируемость системы. Менеджеры могут запускаться или останавливаться во время выполнения проекта без необходимости перезапуска всей системы или отдельных ее компонентов.

## 1.4 Событийно-ориентированный подход

Система SCAD CC является событийно-ориентированной системой, что означает, что обработка данных и информационный обмен между отдельными процессами (менеджерами), в основном, осуществляются только при наступлении каких-либо событий, например, при изменении значений. При отсутствии изменений значений информационный обмен между менеджерами и/или обработка данных не осуществляются.

Система переходит в активное состояние только «по требованию», в связи с чем обеспечивается высокая эффективность использования имеющихся вычислительных ресурсов. Разработчикам приложений доступны функции системы, обеспечивающие возможность подписки пользовательского модуля или интерфейса (потребителя информации) на уведомления об изменениях значений на стороне источника информации (поставщика) (что соответствует модели «поставщик–потребитель»). После успешной активации подписки на изменения, каждое новое значение автоматически передается от источника данных потребителю для дальнейшей обработки.

Связь между менеджерами осуществляется по протоколу TCP/IP. Данный надежный способ связи используется и в пределах компьютера и его операционной системы, и для связи с другими системами. Благодаря использованию стандартного протокола TCP/IP гарантируется надежность, совместимость и производительность.

В событийно-ориентированных системах, как правило, задействуется меньше системных ресурсов, чем в циклических системах такого же масштаба. При одинаковом потоке данных, математически, в событийно-ориентированной системе обеспечивается лучшая воспроизводимость реальных значений процессов. При этом количество ошибок также значительно ниже, чем в циклических системах. Событийно-ориентированные системы также лучше подходят для решения задач, требующих наличия достоверной доказательной базы.

## 1.5 Резервирование, распределенные системы

SCAD CC обеспечивает возможность запуска серверных процессов в режиме горячего резервирования, реализуемого с использованием двух компьютеров. Горячий резерв – это аппаратно независимое решение для обеспечения высокой степени доступности. Концепция горячего резерва использует два сервера, соединенных между собой. Оба сервера работают постоянно, и к ним предъявляются одинаковые функциональные требования (тем не менее, только один сервер всегда является активным; второй сервер синхронизирует данные во время работы с первичным объектом). Если активный сервер выходит из строя, система «на лету» переключается на пассивный сервер, который берет управление на себя и становится активным сервером. В результате всегда обеспечивается доступ к данным или функциям.

Также имеется возможность объединения нескольких автономных систем SCAD CC в одну общую систему (так называемую «распределенную систему»). Каждую подсистему распределенной системы можно сконфигурировать как систему с отдельной станцией или с несколькими станциями, причем она может быть как резервированной, так и нерезервированной. Подсистема в данном соединении означает сервер, на котором работает менеджер событий (это значит, что нет необходимости в целом проекте). В резервированной системе оба работающих резервированных сервера рассматриваются как одна система.

Каждая система может обрабатывать и отображать данные (значения и алармы) других систем. Также можно получить доступ к текущим значениям, алармам и истории каждой системы с помощью распределенных систем, имеющихся в SCAD CC («распределенная база данных»). Когда вы получаете доступ к удаленным элементам точек данных, точки данных подсистем не копируются в локальную систему. Это значит, что точки данных не множатся, а получить доступ к системе можно только в том случае, когда существует соединение с этой конкретной системой.

Менеджер распределенной архитектуры (Dist manager) создает интерфейс для работы с другими системами. Только менеджер распределенной архитектуры отвечает за обмен информацией с подсистемами с заданной конфигурацией. Таким образом менеджер распределенной архитектуры необходимо добавить в консоль и запустить, чтобы обеспечить надлежащее функционирование распределенных систем в SCAD CC.

## 1.6 Пользовательские интерфейсы

В системе присутствуют различные интерфейсы для взаимодействия пользователей с системой.

Панель администрирования проектов (рисунок 2) и консоль (рисунок 3) представляют собой инструменты для управления проектами SCAD CC. С помощью панели администрирования проектов осуществляется создание и администрирование всех проектов на данном компьютере. Консоль используется для запуска конкретного проекта и его мониторинга. Панель администрирования проектов и консоль, как правило, недоступны конечным пользователям. Средство просмотра журналов (модуль вывода системных сообщений менеджеров) ориентировано на инженеров и администраторов, а не на обычных пользователей.

После открытия панели администрирования проектов и подачи команды на запуск выбранного проекта, автоматически открываются консоль и средство просмотра журналов (рисунок 4).

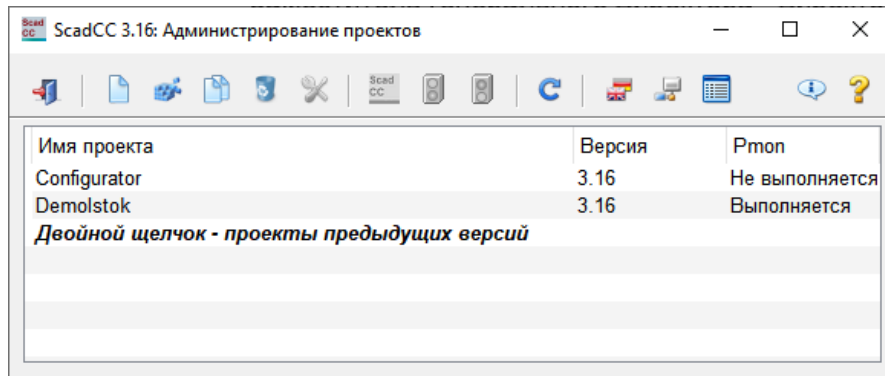


Рисунок 2 – Панель администрирования

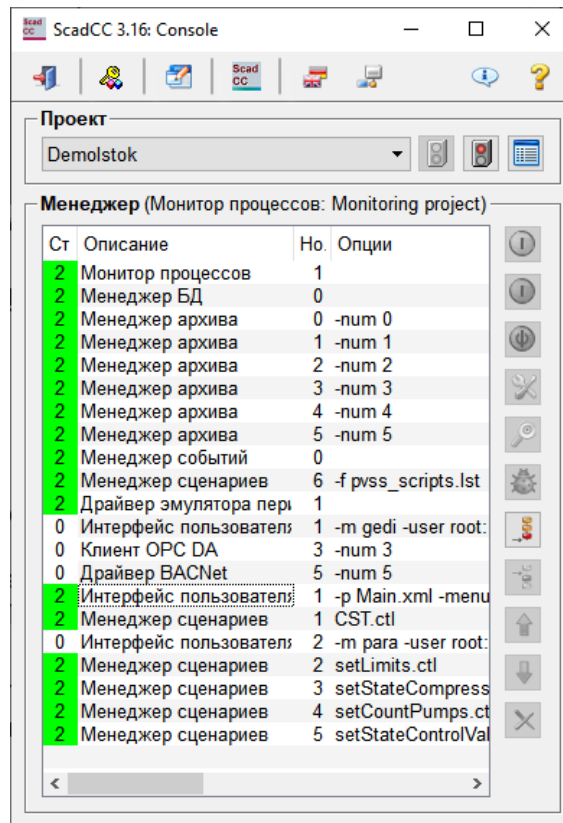


Рисунок 3 – Панель консоли проекта

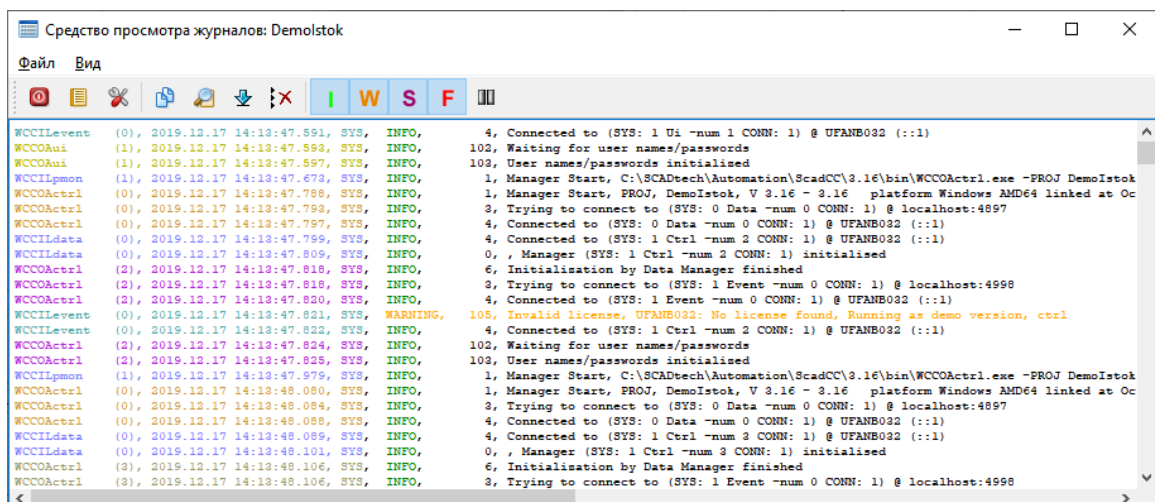
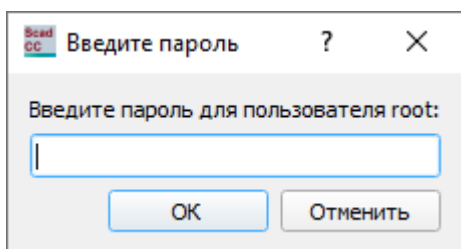


Рисунок 4 – Средство просмотра журналов

В связи с имеющимися в системе функциями обеспечения безопасности отображаются, например, следующие диалоговые окна:



После открытия графического редактора GEDI (рисунок 5) в нем могут быть созданы различные дополнительные графические пользовательские интерфейсы, например, динамические экранные формы процессов, примитивы, динамические информационные диалоговые окна и т.п.

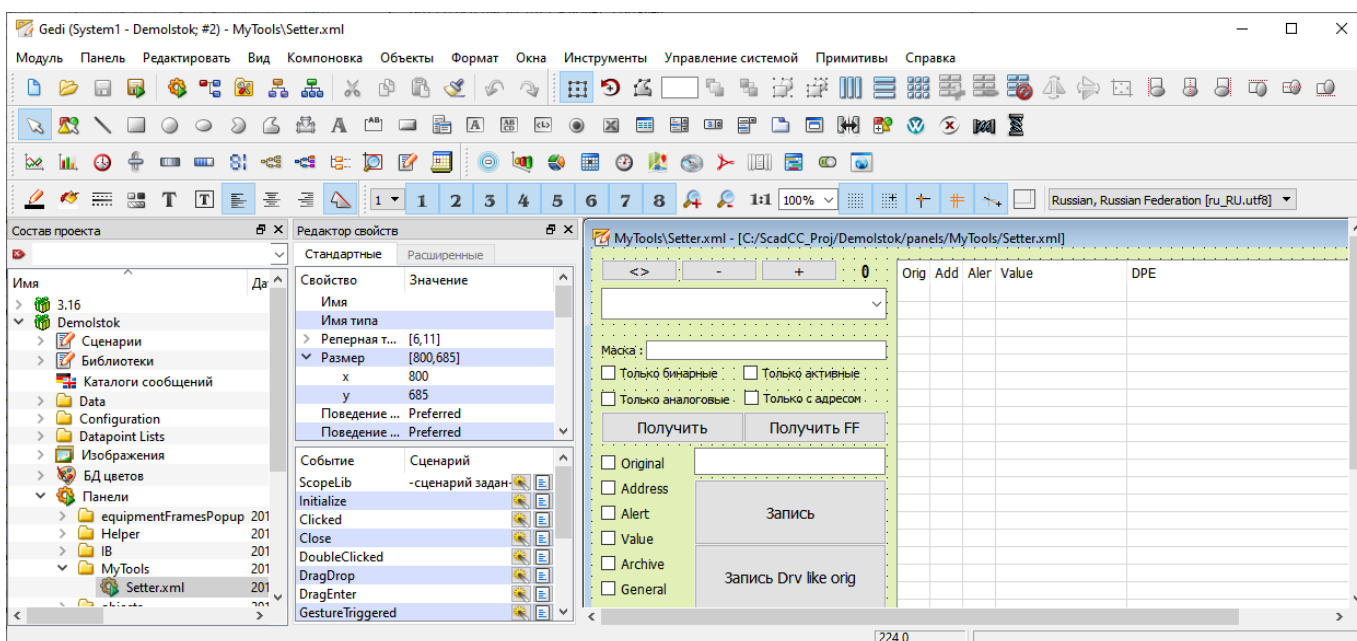


Рисунок 5 – Графический редактор GEDI – инструмент для создания экранных форм, графических объектов и диалогов

Конечному пользователю приложения оперативного управления объектом доступен только исполняемый модуль пользовательского интерфейса VISION. При помощи данного модуля пользователь может осуществлять мониторинг технологических процессов, а также управление технологическими процессами, включая изменение параметров процессов.

Из графического редактора GEDI могут быть открыты редактор PARA (рисунок 6) и панель управления системой (рисунок 7). В модуле PARA создаются и редактируются точки данных (переменные процесса). Панель управления системой представляет собой интерфейс для управления настройками конкретного проекта. Панель управления системой позволяет также вызывать различные дополнительные инструменты для администрирования и диагностики.

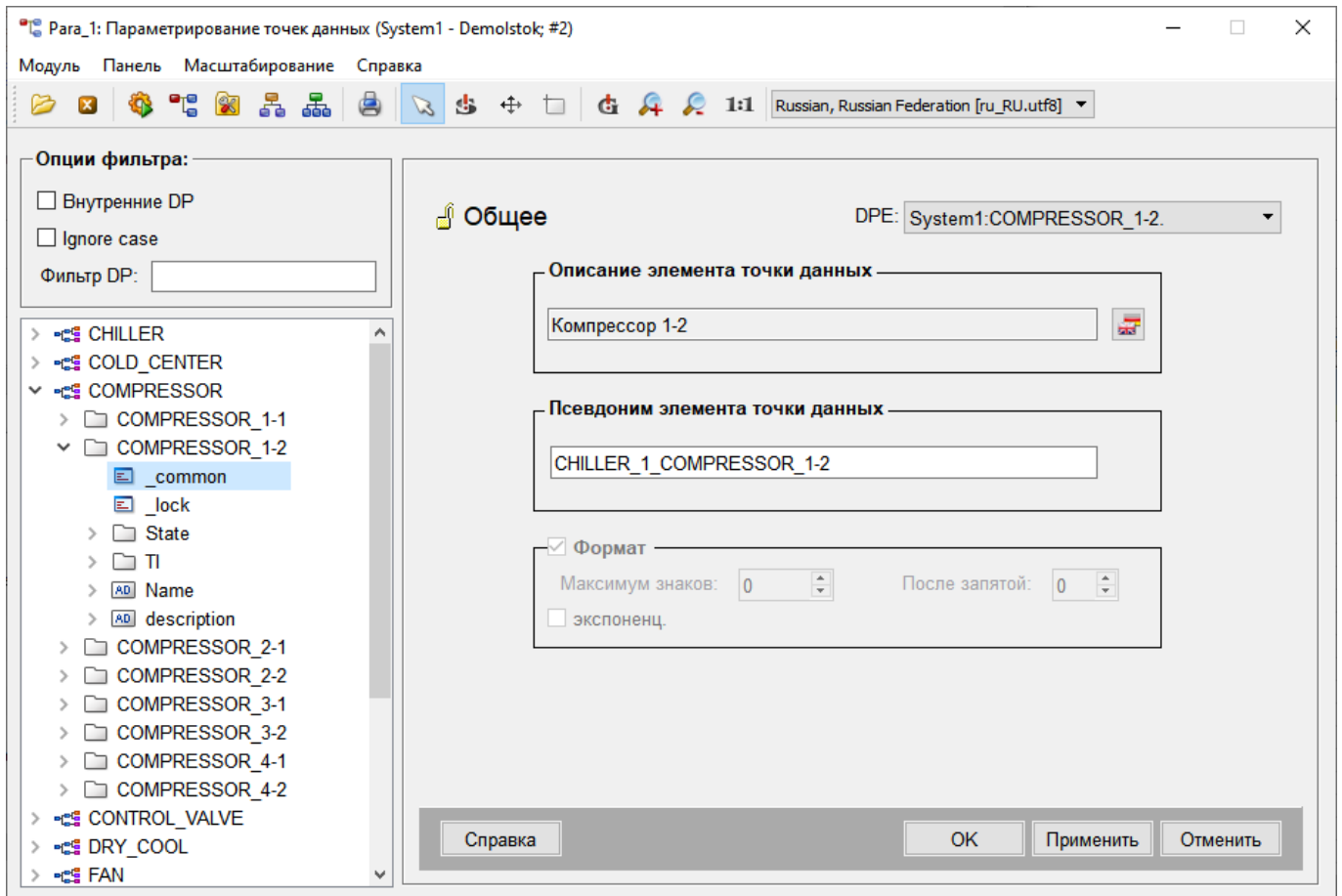


Рисунок 6 – Редактор БД PARA

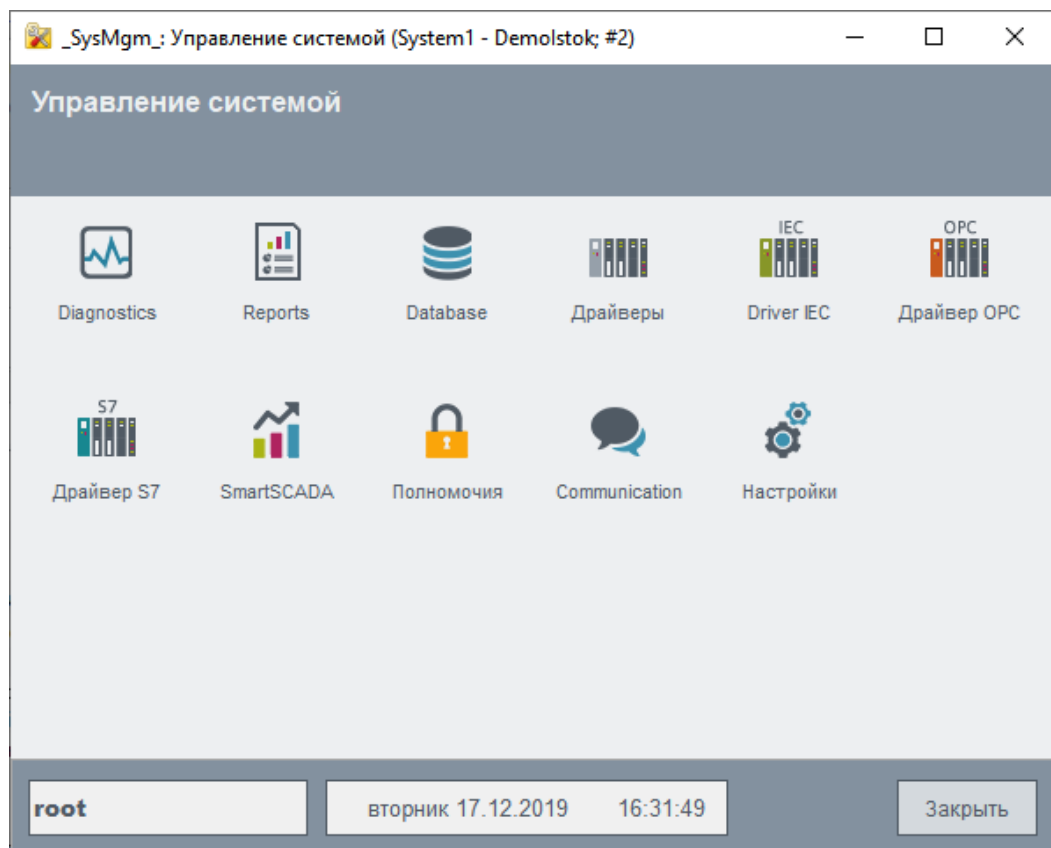


Рисунок 7 – Панель управления системой



## 1.7 Концепция точек данных, образ процесса

Для мониторинга и управления процессом необходим доступ к переменным процесса. Набор используемых переменных должен отражать текущее состояние и значения уставок управляемой установки.

В системе SCAD CC подобные переменные образа процесса называются элементами точек данных. Для обозначения переменных образа процесса также могут использоваться другие термины, например теги, переменные процесса (ПП), технологические переменные, элементы, точки, точки ввода-вывода и т.д.

Тогда как в традиционных SCADA-системах отдельные переменные процесса обрабатываются отдельно, в системе SCAD CC используется более современный подход, при котором практически вся информация о процессе представляется в виде наборов логически связанных переменных, соответствующих отдельным устройствам.

Практический опыт показывает, что количество переменных, объединяемых в рамках одного устройства, обычно колеблется в диапазоне от 4 до 30. При работе с интеллектуальными устройствами, такими как контроллеры, функциональные модули или роботы, количество переменных может существенно возрастать.

В системе SCAD CC для работы с логически связанными значениями используются не разобщенные переменные, а структурированные точки данных, соответствующие конкретным устройствам. Точки данных представляют собой древовидные структуры с возможностью создания произвольного количества подуровней.

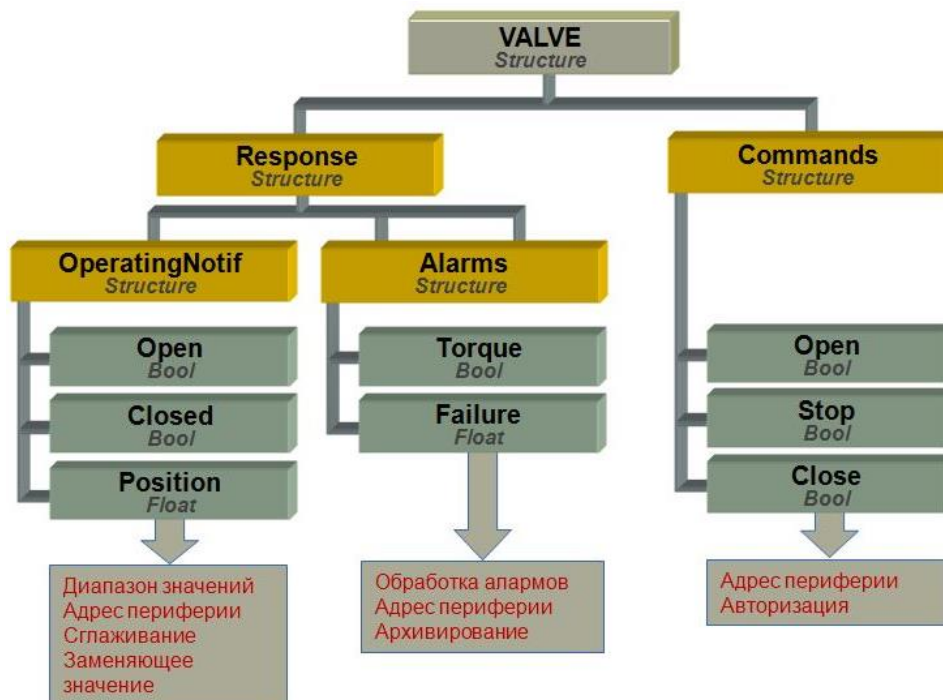


Рисунок 8 – Структурированная точка данных

Полученные значения текущих параметров процесса сохраняются в элементах точек данных (оконечных элементах приведенной древовидной структуры). Одна переменная процесса соответствует одному элементу точки данных. Элемент точки данных является частью структуры точки данных. Для наглядного представления информации в подобную древовидную структуру может быть добавлено любое количество узлов.

Адресация каждого элемента точки данных осуществляется по его полному идентификатору в рамках структуры точки данных. На приведенном выше рисунке адресация элемента точки данных ("Open"), соответствующего сигналу обратной связи об открытии устройства, может выглядеть следующим образом:

`Valve.Response.OperatingNotif.Open`

Дополнительно к возможности именованья/адресации и хранения значений, точки данных позволяют также хранить специальную дополнительную информацию, определяющую их функциональное поведение, например, сведения о допустимом диапазоне значений, настройки обработки алармов, формулы для пересчета значений и т.п. Подобная функциональность может быть определена для каждого элемента точки данных. В SCAD CC подобные носители специальной дополнительной информации, определяющей функциональное поведение элементов точек данных, называются конфигурационными элементами. Создаваться и настраиваться должны только те конфигурационные элементы, которые необходимы для конкретной точки данных (конкретного элемента точки данных).

Для каждого типа физических устройств (привод, клапан, мешалка, контроллер, датчик охранной сигнализации и т.д.) может быть определен отдельный тип точек данных. Структура точки данных, предназначенной для отображения какого-либо физического устройства, наследуется от структуры соответствующего типа точек данных. Т.е. тип точек данных исполняет роль своеобразного шаблона. В объектно-ориентированном программировании тип точек данных можно было бы назвать «классом», а представление физического устройства (точку данных) – «экземпляром» класса.

Таким образом, создание и настройка множества сигналов процесса, относящегося к определенному типу устройств, сводится к однократной операции. Ранее созданные типы точек данных, соответствующие какому-либо типу устройств (например, типу «счетчик времени наработки»), могут включаться в состав новых типов точек данных (тип в типе), что обеспечивает новые возможности для эффективной разработки.

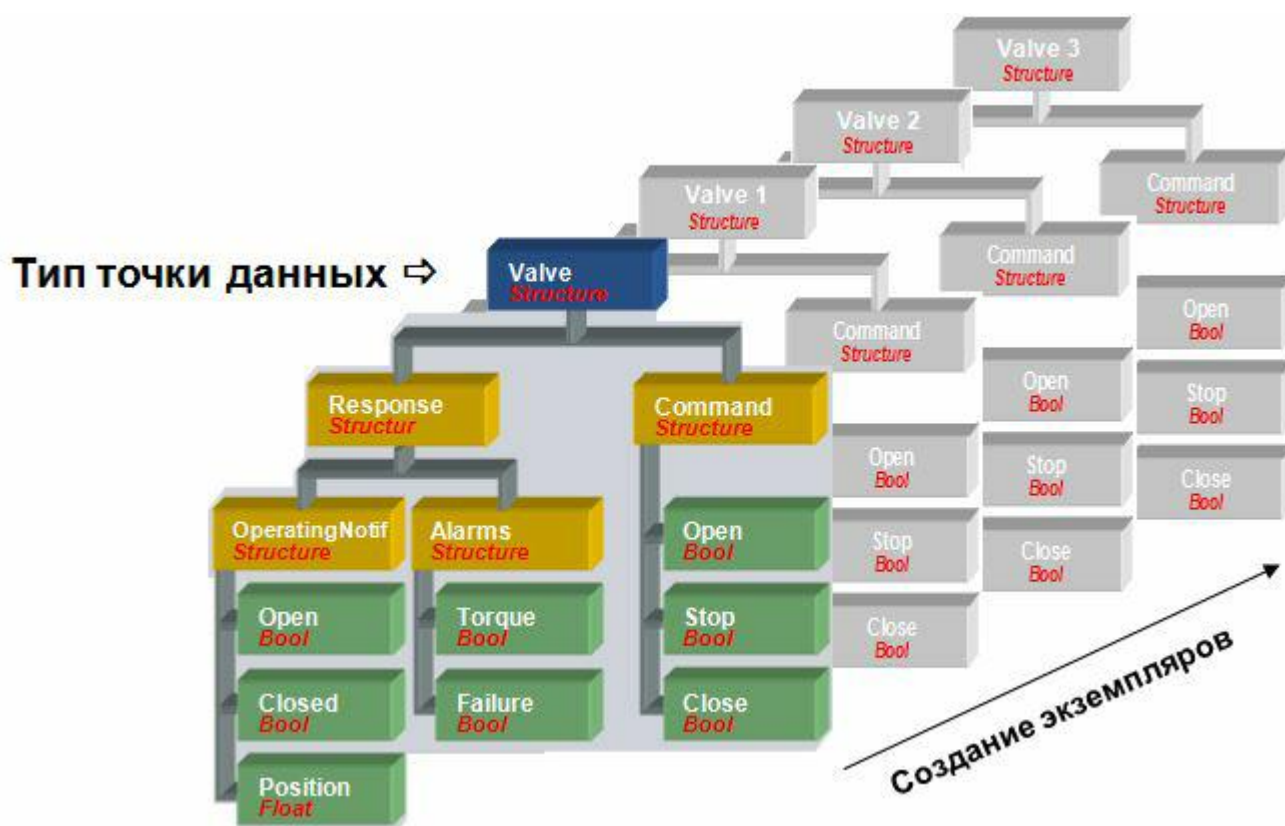


Рисунок 9 – Точки данных клапанов 1 ... 3 как экземпляры типа точек данных "Valve"

## 1.8 Общие сведения о драйверах

Информационный обмен между системой управления на базе SCAD CC и базовыми средствами автоматизации (ПЛК, удаленными оконечными устройствами и т.п.) со стороны системы SCAD CC осуществляется с использованием специальных менеджеров, именуемых «драйверами». Выбор драйверов зависит от протоколов связи и используемых физических сред передачи данных:

- Протоколы последовательной передачи данных: RK512, 3964R, Muehbus и т.д.
- Полевая шина: Profibus DP, Profibus FMS и т.д.
- Ethernet: ModbusTCP (OpenModbus), Industrial Ethernet (S7), Ethernet IP (AB) и т.д.
- Системы телеуправления: SSI (Ethernet), IEC 60870-5-101, IEC 60870-5-104 и т.д.
- Унифицированные интерфейсы: OPC DataAccess, OPC Alarms&Events.

Менеджер драйвера предназначен для приведения формы передачи информации, разработанной определенным производителем (протокол, физическая среда передачи), к форме передачи информации в системе SCAD CC. При этом драйвером автоматически выполняется ряд задач:

- Обеспечение согласования с физической средой передачи данных;
- Декодирование информации, передаваемой по используемому протоколу (анализ кадров/ телеграмм);

- Мониторинг подключения;
- Автоматическое сравнение старого и нового значений (низкоуровневое сглаживание);
- Сглаживание по значению и времени (высокоуровневое сглаживание);
- Преобразование типов данных;
- Преобразование применяемой формы передачи информации в событийно-ориентированный обмен данными.

Менеджер драйвера добавляется в список менеджеров в консоли таким же образом, как и любые другие менеджеры. В последующем, в зависимости от специфики конкретной установки, потребуется индивидуальная настройка каждого драйвера.

Все элементы точек данных, без принятия дополнительных мер для обеспечения взаимодействия с внешними системами, по своей сути являются внутренними: их значения могут быть изменены или прочитаны только менеджерами SCAD CC. При этом обмен данными с базовыми средствами автоматизации (ПЛК, удаленными оконечными устройствами и т.п.) автоматически не обеспечивается.

Для получения элементом точки данных значения из внешнего источника, например ПЛК, соответствующему элементу должен быть присвоен периферийный адрес. Подобное присвоение осуществляется при помощи конфигурационного элемента. Для присвоения элементу точки данных периферийного адреса необходимо выполнить следующие действия в редакторе PARA:


- 1) Добавить к элементу точки данных соответствующий конфигурационный элемент или шаблонный конфигурационный элемент "\_address".
- 2) Выбрать необходимый драйвер.
- 3) Указать аппаратный или сетевой адрес (например, '%MW1021' для Modbus TCP при применении ПЛК Schneider Premium или Quantum).
- 4) Указать направление информационного обмена и, для некоторых протоколов, типы доступа (ввод, вывод, ввод/вывод).


В системе SCAD CC поддерживается одновременный запуск нескольких драйверов. Данные драйверы могут принадлежать как к одному, так и к различным типам. Например, система SCAD CC может одновременно взаимодействовать с контроллером Simatic по протоколу S7, с системой телеуправления по протоколу IEC 60870-5-104 и с сервером OPC по протоколу OPC DA.

## 2 Модель данных

### 2.1 Точки данных как носители информации

С целью обеспечения возможности создания в системе SCAD CC пользовательских интерфейсов, позволяющих отображать имеющуюся информацию, в системе требуется наличие носителей данной информации. Подобными носителями информации являются переменные, значения которых отражают текущие параметры процесса. Переменные процесса в системе SCAD CC называются элементами точек данных.

Для отображения существующих точек данных, а также для создания новых точек данных, как правило, используется редактор базы данных PARA. Редактор PARA может быть открыт, например, из уже открытого графического редактора GEDI. Для этого используется кнопка .

Редактор PARA может быть также запущен при помощи консоли с использованием отдельного менеджера пользовательского интерфейса. Для этого в консоли нажав кнопку  откройте диалоговое окно для выбора менеджера, выберите менеджер «WCCOAni» и введите в поле «Опции» строку "-m para" (без кавычек). Задайте режим запуска «вручную».

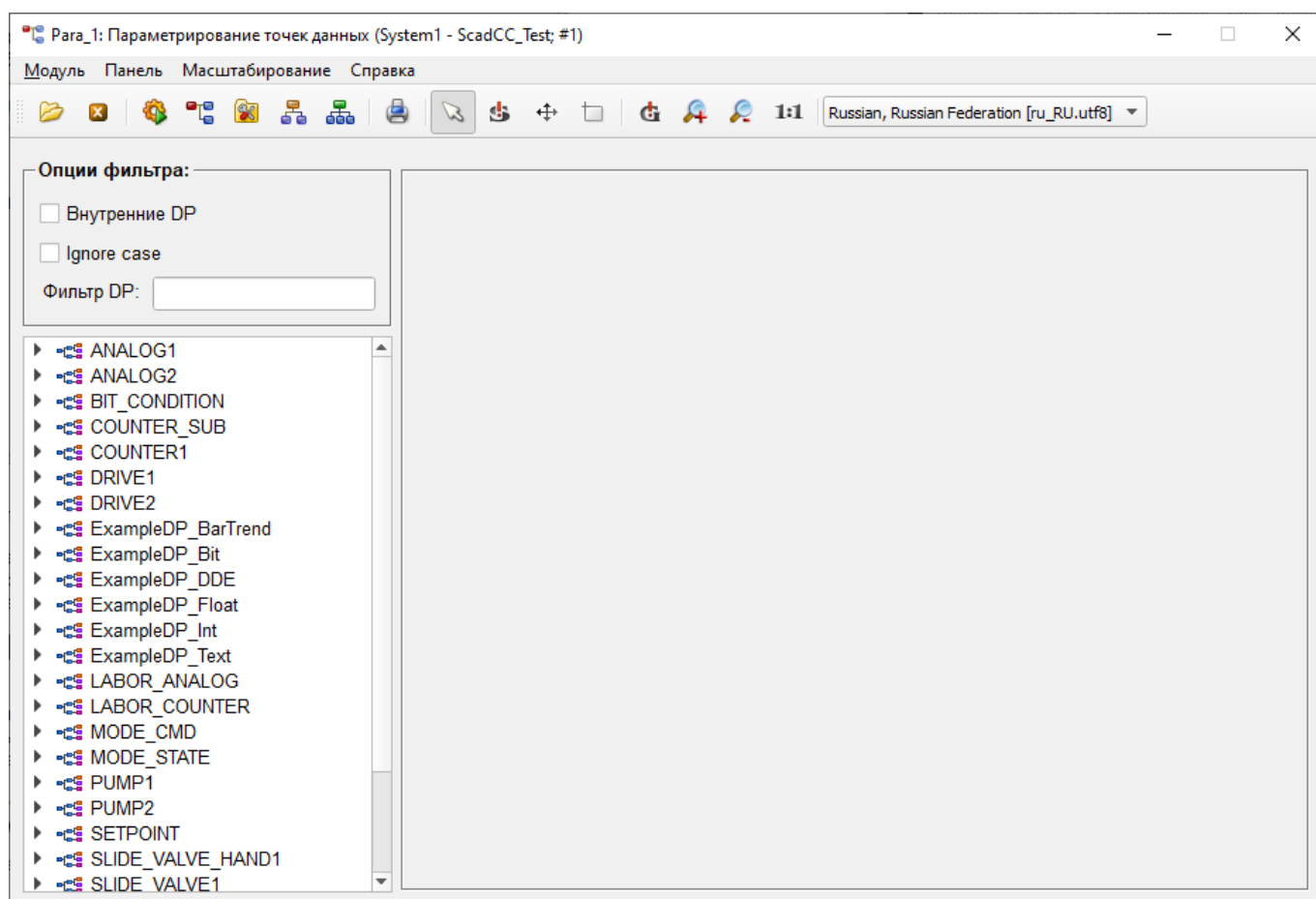


Рисунок 10 – Редактор базы данных PARA

В левой части экрана (см рисунок выше) выводится список типов точек данных, доступных в рамках текущего проекта. Каждый из этих типов соответствует отдельному классу устройств или набору логических связанных переменных. При нажатии на знак ▶ перед названием типа точек данных открывается список точек данных (экземпляров, устройств) данного типа.

Тип точек данных представляет собой своего рода шаблон структуры для точек данных этого типа. При создании типа задаются его структура, имя и некоторые другие настройки.

Каждая точка данных может отображать в системе SCAD CC реальное физическое устройство или набор логически связанных переменных. Точки данных состоят из одного или нескольких элементов точек данных, объединяемых в практически произвольную структуру, изначально задаваемую при создании соответствующего типа точек данных. Каждая точка данных соответствует какому-либо типу точек данных, что соответствующим образом отображается в редакторе PARA. Список точек данных интересующего типа открывается при нажатии на знак ▶ перед названием типа точки данных или при двойном щелчке левой кнопкой мыши по этому типу точек данных.

На рисунках ниже представлены два типа точек данных, создаваемых по умолчанию в новом проекте, а также соответствующие точки данных (экземпляры). В первом примере типу точек данных "ExampleDP\_Float" соответствуют указанные точки данных (экземпляры). Каждая точка данных этого типа состоит только из одного элемента точки данных, и поэтому может отображать только одну переменную процесса. Подобный подход используется во многих SCADA-системах, но является, скорее, исключением в системах SCAD CC.

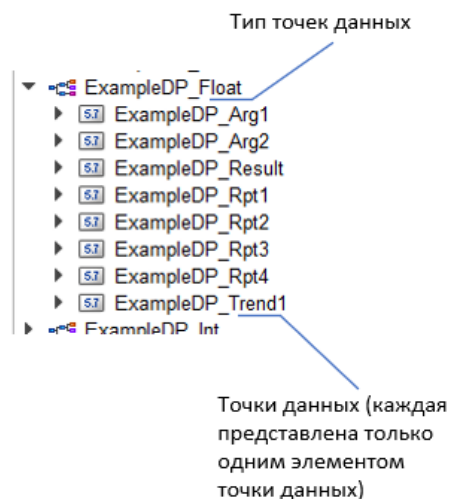


Рисунок 11 – Экземпляры типа точек данных "ExampleDP\_Float"

Тип точек данных "PUMP1" (на рисунке ниже) больше соответствует структурам данных, обычно используемым в SCAD CC. Устройство-ориентированная точка данных является носителем множества переменных процесса. Непосредственно после создания нового проекта, типу "PUMP1" соответствует только один экземпляр, также представляющий собой исключение:

данный экземпляр является мастер-точкой данных (MP), которая служит для хранения шаблонных настроек для данного типа точек данных.



Рисунок 12 – Экземпляры типа точек данных "PUMP1"

Каждый элемент точки данных содержит значение переменной процесса, а также основные атрибуты значения. В связи с этим элемент точки данных является носителем части образа процесса в системе.

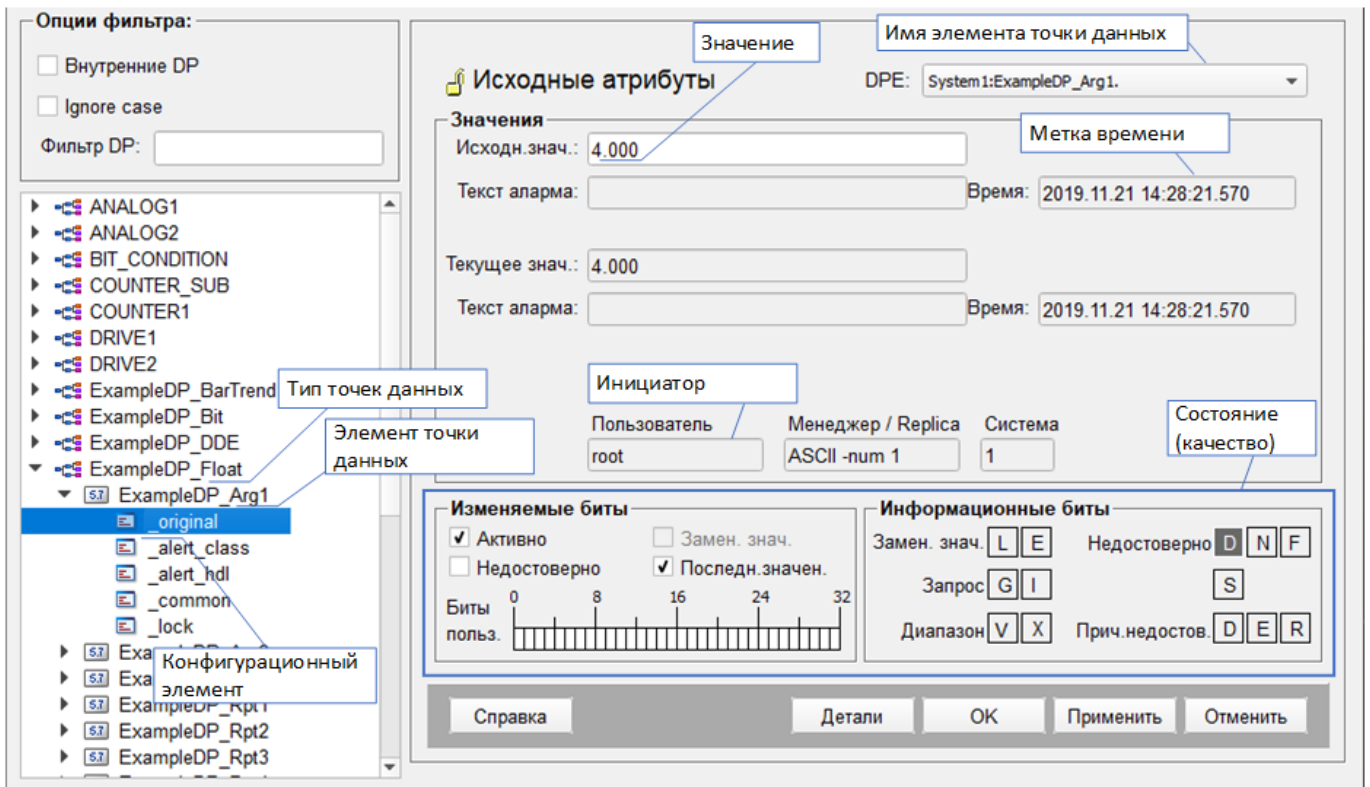


Рисунок 13 – Тип точек данных "ExampleDP\_Float" с точкой данных "ExampleDP\_Arg1"

Атрибуты также отображаются в редакторе базы данных PARA. Кроме этого, существуют два различных поля для хранения значения:

<b>Исходное значение</b>	В атрибуте «Исходное значение» элемента точки данных хранится изначально полученное (например, от ПЛК) значение. Данное значение доступно всегда и может быть опрошено даже после его автоматической обработки, например, после применения заменяющего значения. При получении, например, от полевого устройства нового значения сначала перезаписывается именно исходное значение
<b>Текущее значение</b>	Все функциональные модули SCAD CC используют именно это значение. Как правило, оно представляет собой копию исходного значения. Однако, если исходное значение признается недостоверным, используется Заменяющее значение. При чтении значения элемента точки данных (для вывода или обработки) всегда считывается текущее значение

Элементы точек данных могут хранить данные различных типов. Помимо стандартных двоичных и аналоговых переменных доступны, в том числе, массивы и структуры. В связи с этим точки данных могут использоваться не только для отображения состояния процесса, но и для выполнения административных задач, вычислений, работы с рецептами и т.п.

Тип данных	Разрядность	Значение
bool	1 бит	Двоичная переменная
int	32 бита	Целое число
unsigned	32 бита	Целое число без знака
float	64 бита	Число с плавающей точкой
bit32	32 бита	Набор битов
char	8 бит	Символ
string	n x 8 бит	Строка (длиной n символов)
langString	m x n x 8 бит	Многоязычная строка (n языков)
time	48 бит	Время / продолжительность
dpid	-	Идентификатор точки данных
typeref	-	Ссылка на тип
blob	произвольная	Двоичная структура ( <b>binary large object</b> – большой двоичный объект)
dyn_... (dyn_string,...)		Динамические массивы, в которых могут храниться данные любого из перечисленных выше типов (размер и резервируемое для хранения информации пространство могут динамически изменяться для каждого элемента массива)
array... (char array,...)		Статические массивы, в которых могут храниться данные любого из перечисленных выше типов (количество элементов массива определяется изначально при настройке).

Тип данных элемента точки данных определяется изначально при создании типа точки данных (вместе со схемой идентификации/адресации). Возможное последующее изменение типа



данных (на уровне типа точек данных) допустимо, но приведет к изменению типа данных всех соответствующих элементов точек данных.

## 2.2 Адресация

Обращение к какому-либо отдельному элементу точки данных в системе SCAD CC может быть выполнено с использованием полного идентификатора (имени) элемента точки данных. В модуле PARA имя элемента точки данных отображается в поле «DPE» справа сверху (см. рисунок в предыдущем разделе). Формат имени элемента точки данных:

ИмяСистемы:ИмяТочкиДанных.Элемент1.Элемент2...ЭлементN

Имя системы может быть изменено, например, в случае развертывания «распределенной системы» SCAD CC, состоящей из нескольких взаимодействующих друг с другом отдельных систем SCAD CC.

Адресация какой-либо переменной процесса может выполняться в системе SCAD CC двумя различными способами. Первым способом является адресация с использованием идентификатора (имени) элемента точки данных в точечной нотации.

Вторым способом является адресация с использованием псевдонимов. Псевдонимы могут применяться для создания альтернативной интуитивно понятной системы адресации в рамках отдельной установки. В отличие от идентификаторов (имен) элементов точек данных, псевдонимы не обязательно должны быть организованы в виде иерархической структуры.

## 2.3 Конфигурационные элементы и мастер-точки данных

Элементы точек данных системы SCAD CC являются носителями не только значений, единиц измерения, описаний и т.п., но и других данных. Часть этих данных используется в информационных целях. Другая часть этих данных предназначена для настройки функциональности элементов точек данных, например, алгоритмов преобразования значений, правил формирования алармов и т.д.

Носители подобных настроек входят в состав элементов точек данных и называются в системе SCAD CC конфигурационными элементами.

С целью обеспечения консистентности в ходе настройки всех экземпляров точек данных одного типа используются мастер-точки данных. Конфигурационные элементы, создаваемые на уровне мастер-точки данных, будут в дальнейшем «копироваться» в создаваемые точки данных. Структура создаваемых точек данных, именование элементов структуры и их типы данных задаются в типе точек данных. Настройки, выполняемые на уровне мастер-точки данных, могут быть объявлены либо фиксированными, и в таком случае они будут применяться ко всем создаваемым точкам данных без изменений, либо могут быть объявлены изменяемыми, и в таком случае на уровне создаваемых точек данных будет возможна их дополнительная корректировка.

Таким образом, на уровне мастер-точки данных возможно применение различных типов конфигурационных элементов. Конфигурационные элементы, применяемые на уровне мастер-точек данных, именуются «шаблонными конфигурационными элементами».

Без использования мастер-точек данных SCAD CC предоставляет возможность индивидуальной настройки создаваемых точек данных. С использованием мастер-точек данных дополнительно появляется возможность предварительного создания набора настроек, который будет применяться к создаваемым точкам данных. При удалении мастер-точки данных существующего типа точки данных, ранее созданные на основе удаленной мастер-точки, продолжают работать, однако утрачивают возможность наследования функциональности удаленной мастер-точки.

### 3 Создание и настройка точек данных

#### 3.1 Создание типов точек данных

Для создания типа точек данных перейдите в редактор PARA. В области древовидной структуры щелкните ПКМ по пустому белому пространству и выберите в открывшемся контекстном меню элемент [Создать тип точек данных].

После этого автоматически откроется окно редактора типов точек данных, в котором выполняется разработка структуры. Далее:

- 1) Замените предлагаемое по умолчанию имя типа точек данных "newDpType".

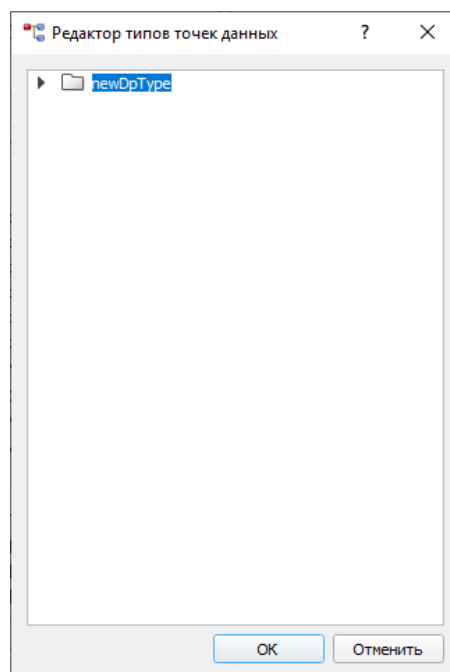


Рисунок 14 – Создание типа точек данных в редакторе модуля PARA

- 2) Щелкните ПКМ по созданному типу точки данных и выберите в открывшемся контекстном меню элемент [Добавить узел].
- 3) Измените имя добавленного узла, например, на "state".

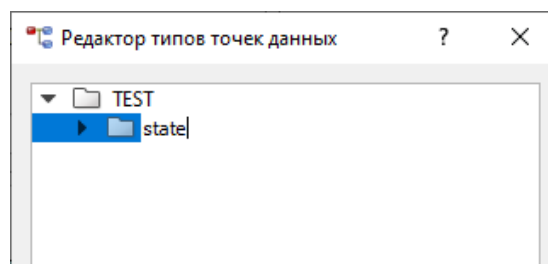


Рисунок 15 – Добавление узла

- 4) Добавьте к созданному узлу ("state") подузел, например, "on" (см. шаги 2 и 3 выше).

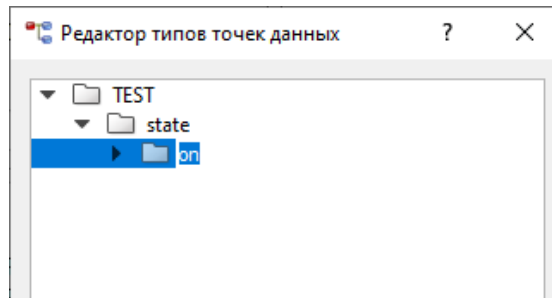


Рисунок 16 – Добавление подузла

5) Измените тип данных для узла ("on"), например, на "bool". Для этого щелкните ПКМ по узлу ("on") и выберите в открывшемся контекстном меню элемент [Тип элемента] → [bool].

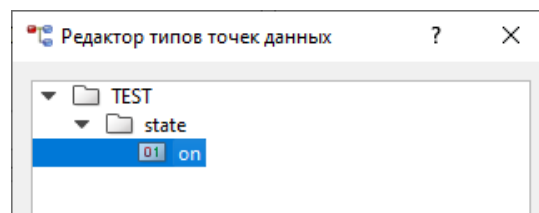



Рисунок 17 – Назначение типа данных

Узлы структуры обозначаются значком папки. Конечные элементы точек данных обозначаются значками, соответствующими их типам (bool, int, float...). Например, тип данных "bool" обозначается значком .

В этом же редакторе возможны удаление узлов, их переименование и другие операции.

После ввода в редакторе типов точек данных всей целевой структуры нажмите кнопку [OK]. Созданный тип точки данных отобразится в древовидной структуре.

### 3.2 Создание точек данных

Типы точек данных не предназначены для хранения какой-либо информации о процессе. Они являются своего рода шаблонами структур для точек данных, отражающих в системе SCAD CC реальные физические устройства. Новые точки данных могут быть созданы с использованием контекстного меню, открывающегося при щелчке ПКМ по типу точек данных в древовидной структуре в редакторе PARA. В контекстном меню необходимо выбрать пункт [Создать точку данных], далее в открывшемся окне нужно ввести имя точки данных и нажать кнопку [OK].

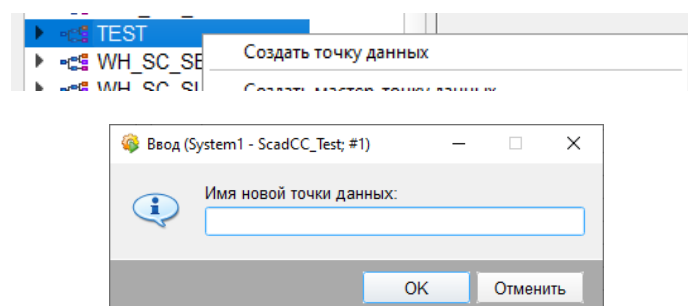


Рисунок 18 – Создание точки данных с использованием контекстного меню

После этого под выбранным типом может быть открыта древовидная структура, в которую будет входить только что созданная точка данных.

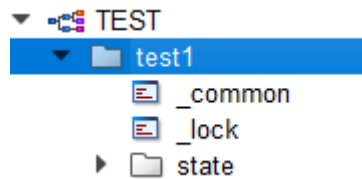


Рисунок 19 – Созданная точка данных

Ограничения на количество точек данных, которые могут быть созданы на основе одного типа, практически отсутствуют. Имена точек данных должны быть уникальными в рамках всей системы. Кроме этого, имена точек данных могут содержать только буквы алфавита, цифры от 0 до 9 и знак подчеркивания «\_». Имя точки данных должно начинаться либо с буквы, либо со знака подчеркивания. Использование цифр допускается со второй позиции. SCAD CC чувствительна к регистру, например, строка "AAA" не идентична строке "aaa".

Для создания точек данных должен существовать как минимум соответствующий тип точек данных. При отсутствии мастер-точки данных создаваемые точки данных наследуют от типа точек данных только структуру, имена элементов и типы данных.

При наличии мастер-точки данных создаваемые точки данных содержат также конфигурационные элементы. Определение атрибутов конфигурационных элементов, которые затем должны настраиваться отдельно на уровне каждой создаваемой точки данных, выполняется при добавлении шаблонного конфигурационного элемента к мастер-точке данных.

В системе SCAD CC существует возможность одновременного создания сразу нескольких точек данных. Для этого:

- 1) Щелкните ПКМ по нужному типу и выберите в открывшемся контекстном меню элемент [Создать набор точек данных].
- 2) В открывшемся диалоговом окне задайте формат имен создаваемых точек данных. Для этого в поле «Текст до номера» введите префикс имен и задайте диапазон номеров. Для номера будет достаточно одной позиции (поле «Минимальное количество позиций»).
- 3) Поле «Текст после номера» оставьте пустым.
- 4) Для подтверждения введенных настроек нажмите кнопку [ОК].

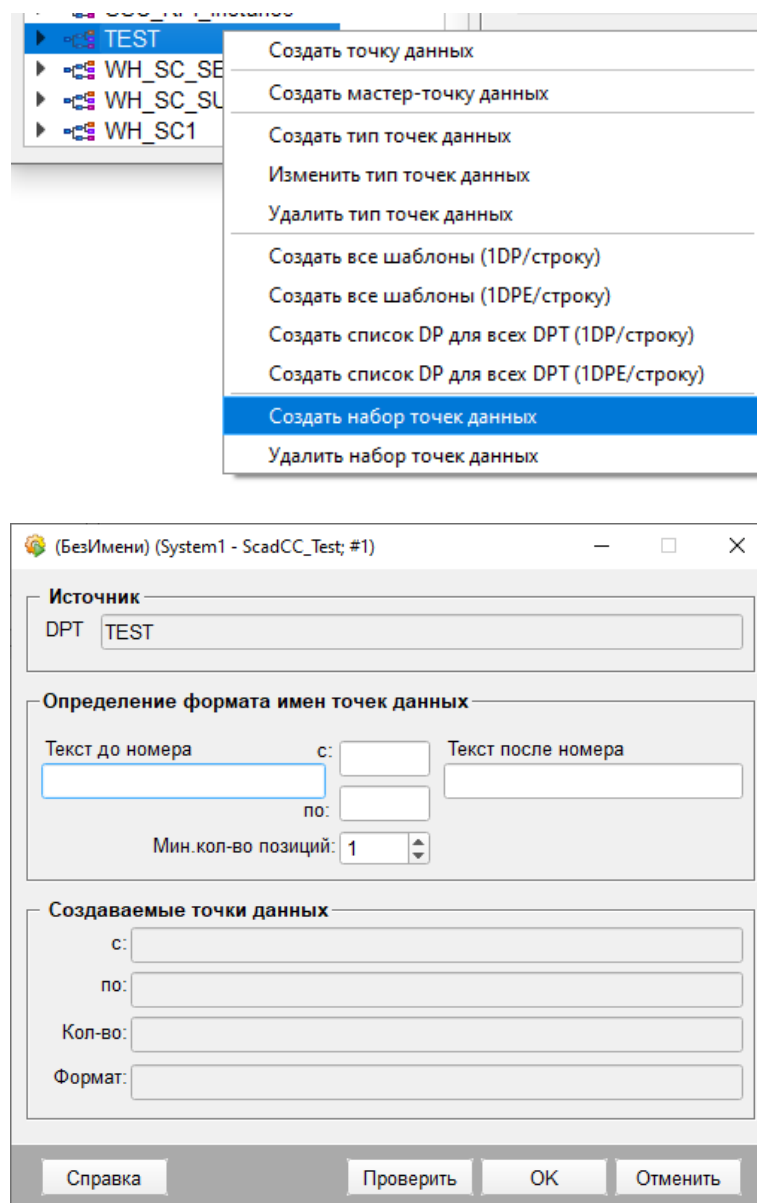


Рисунок 20 – Создание нескольких точек данных в модуле PARA с автоматическим присвоением имен

Редактор базы данных PARA представляет собой эффективный инструмент для создания и настройки точек данных. Однако для создания большого количества точек данных, соответствующих большому количеству устройств, следует использовать механизм импорта списков точек данных при помощи менеджера ASCII системы SCAD CC.

В модуле PARA для удаления точек данных может быть использовано контекстное меню типа точек данных или точки данных. Поддерживается также одновременное удаление набора точек данных.

### 3.3 Создание мастер-точки данных

В системе реализована возможность однократного выполнения настроек для однотипных точек данных вместо индивидуальной настройки конфигурационных элементов каждого элемента

каждой точки данных. Для этого создаются мастер-точки данных. Создать мастер-точку данных можно используя контекстное меню: выберите тип точек данных, щелкните по нему ПКМ и выберите в открывшемся контекстном меню элемент [Создать мастер-точку данных], после чего мастер-точка данных будет создана автоматически. При этом ее имя будет соответствовать имени типа точек данных с добавленным префиксом "\_mp\_" (сокращение от "master data point").

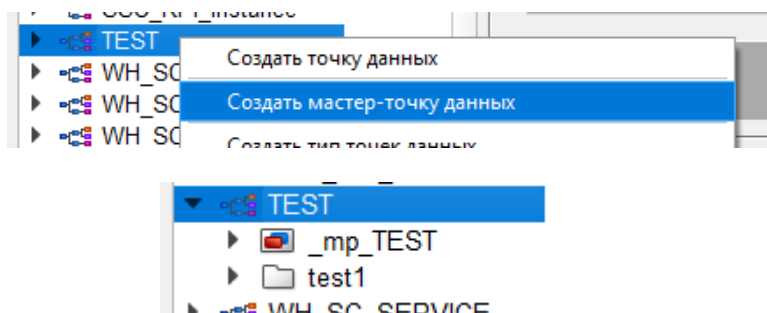


Рисунок 21 – Создание мастер-точки данных для типа точек данных с использованием контекстного меню

С использованием мастер-точек данных дополнительно появляется возможность предварительного создания набора настроек, который будет применяться к создаваемым точкам данных. Для этого предназначена функция добавления шаблонного конфигурационного элемента.

При добавлении шаблонного конфигурационного элемента к мастер-точке данных параметры шаблонных конфигурационных элементов могут быть определены либо как фиксированные, и тогда они будут применяться без каких-либо изменений ко всем создаваемым точкам данных соответствующего типа, либо как изменяемые и настраиваемые индивидуально на уровне каждой новой точки данных в модуле PARA.

### 3.4 Настройка элементов точек данных

В некоторых случаях для элементов точек данных требуется выполнение дополнительных настроек. Например, каждый элемент может хранить следующую информацию:

- Описание;
- Псевдоним (альтернативная система обозначений/адресации, используемая одновременно с именами точек данных);
- Формат (количество знаков всего и после запятой);
- Единицы измерения.

Для открытия диалогового окна настроек для нужного элемента точки данных необходимо щелкнуть ЛКМ по его подчиненному элементу – конфигурационному элементу "\_common".

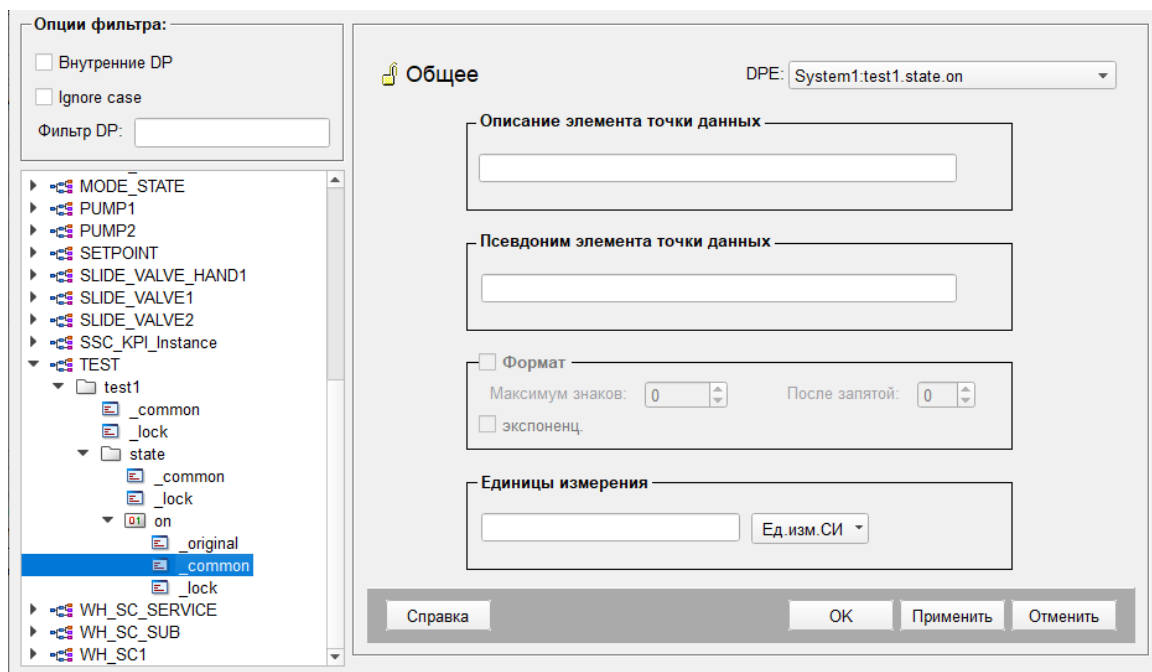


Рисунок 22 – Настройка элементов точек данных


### 3.5 Импорт настроек с использованием ASCII-файлов (массовое параметрирование)


SCAD CC предоставляет возможность экспорта в файлы ASCII типов точек данных, мастер-точек данных и точек данных, включая их настройки. Поддерживается также обратный процесс, т.е. импорт из файлов ASCII во внутреннюю базу данных SCAD CC типов точек данных, мастер-точек данных и точек данных и их настроек. Для выполнения данных операций используется менеджер ASCII.

Менеджер ASCII представляет собой инструмент, запускаемый из командной строки. Менеджер ASCII может запускаться в фоновом режиме и может быть легко интегрирован в системы более высокого уровня. В системе SCAD CC также имеется графический интерфейс для управления менеджером ASCII. Данный интерфейс именуется как «Панель менеджера ASCII». Панель менеджера ASCII может быть открыта из панели управления системой, раздел «База данных».


Импорт настроек проекта с использованием панели менеджера ASCII выполняется в следующем порядке:

1) Скопируйте содержимое директории с файлами импорта в директорию проекта и подтвердите запрос на перезапись существующих файлов.

2) В редакторе GED1 нажмите кнопку  для открытия панели управления системой и щелкните в ней ЛКМ по значку [База данных].

3) Нажмите на кнопку запуска менеджера ASCII .



- 4) В открывшемся менеджере ASCII в области «ASCII-файл» нажмите кнопку  и выберите в открывшемся окне выбора файлов файл импорта из директории ".../dplist/".
- 5) В области «Направление» выберите переключатель [Импорт].
- 6) Установите флажок [Автоматически обновить существующие типы DP].
- 7) Установите флажок [Загрузить данные с обработкой алармов].
- 8) Для начала процесса импорта нажмите кнопку [Начать].
- 9) После удачного завершения импорта закройте панель менеджера ASCII.

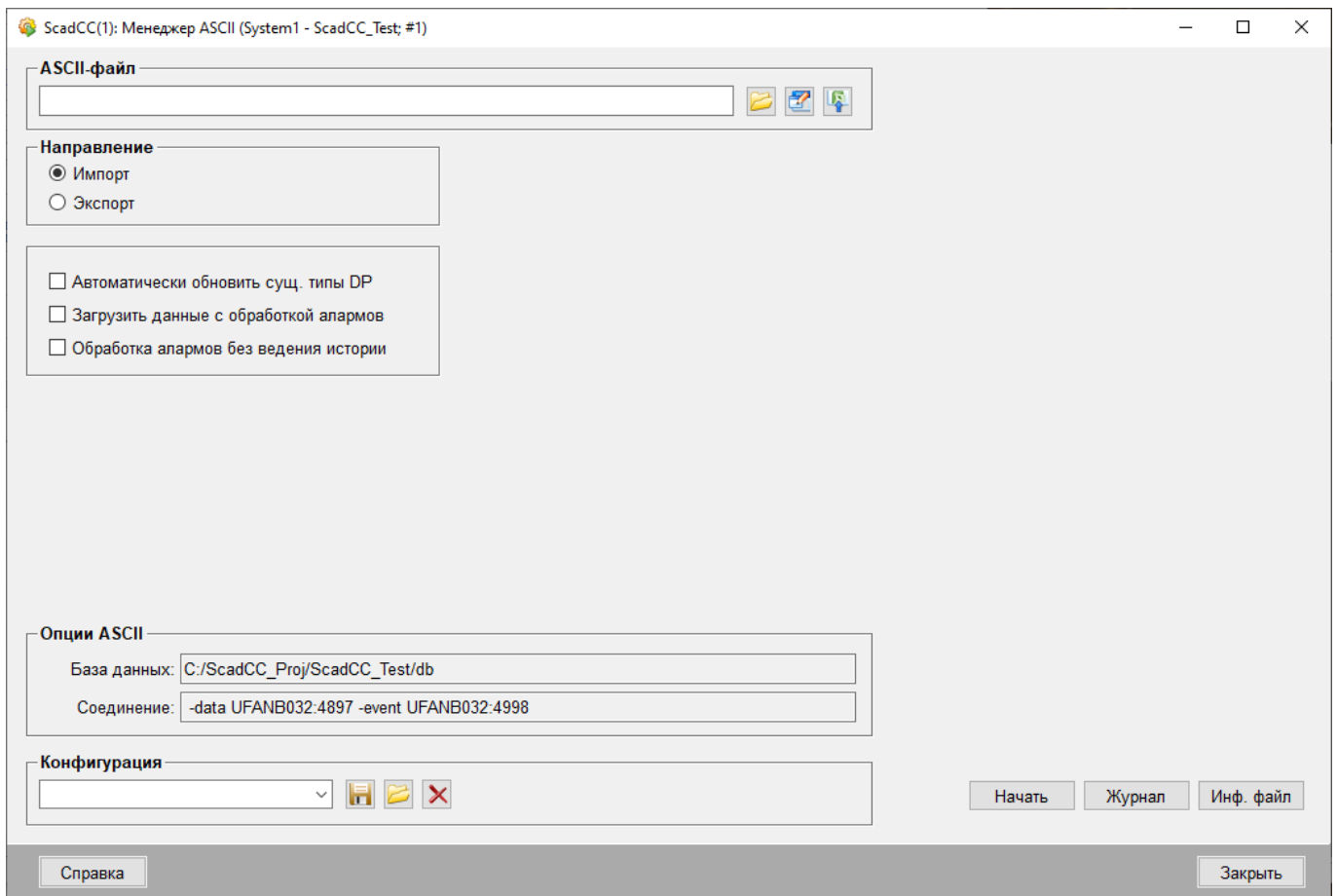


Рисунок 23 – Панель менеджера ASCII (импорт)

SCAD CC поддерживает несколько форматов при импорте и экспорте типов точек данных и точек данных. Данные форматы представляют собой, в основном, форматы ASCII/CSV. В качестве разделителя используется знак табуляции. Данные в этих файлах не кодируются и могут быть просмотрены при помощи любого текстового редактора.

Имеется возможность выбора данных, подлежащих импорту или экспорту. Например, имеется возможность импорта или экспорта только отдельных выбранных точек данных, типов точек данных, отдельных конфигурационных элементов или только изменений, внесенных после указанной даты.

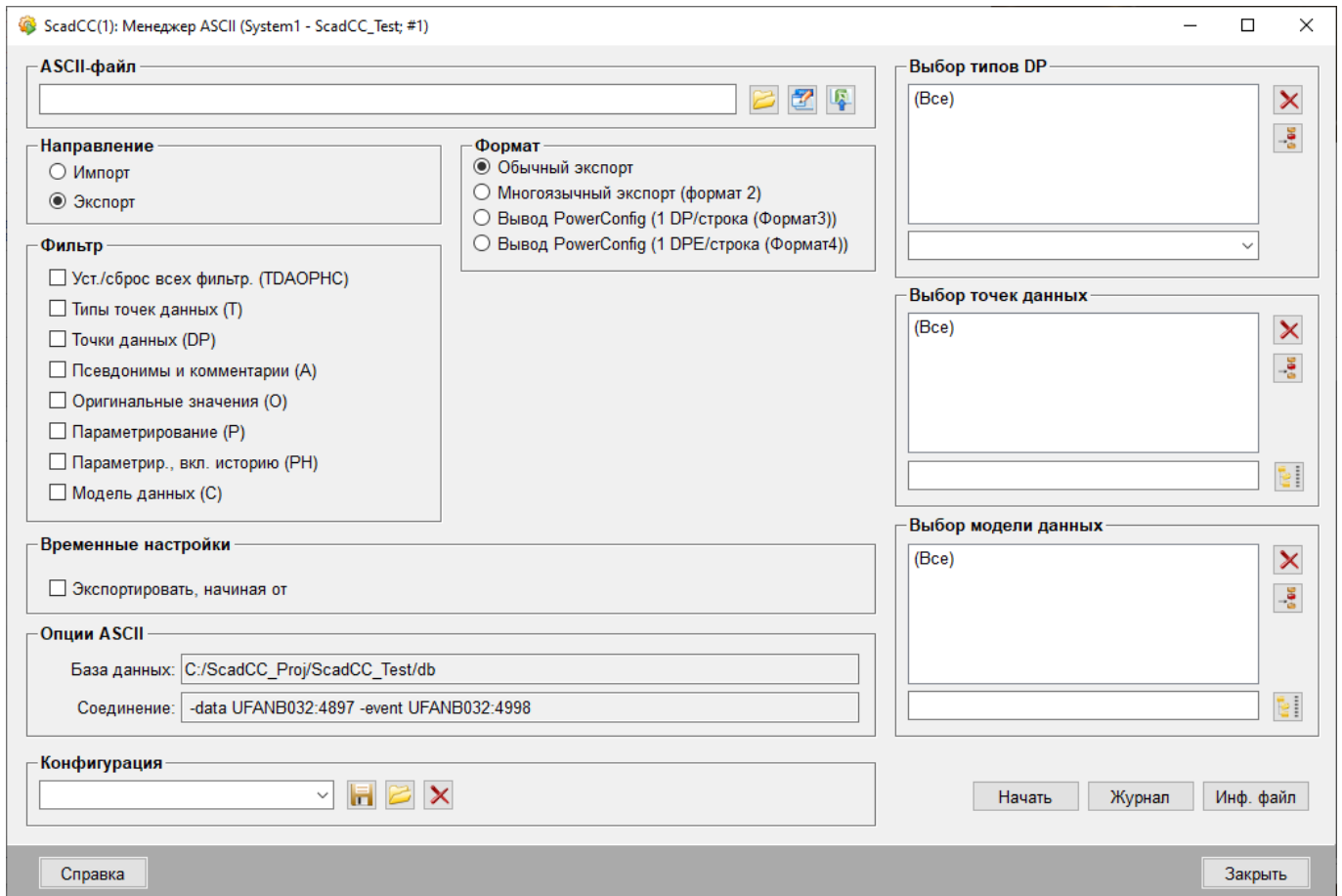


Рисунок 24 – Панель менеджера ASCII (экспорт)

Менеджер ASCII системы SCAD CC может быть использован для создания типов точек данных и точек данных. Для выполнения операции импорта должны быть запущены как минимум менеджер событий, менеджер данных и как минимум один драйвер эмуляции периферии. При этом поддерживается изменение настроек во время выполнения проекта. При выполнении операций необходимо учитывать дополнительную нагрузку на системные ресурсы (процессор, память). Экспорт данных может осуществляться также и при остановленной системе.

Форматы менеджера ASCII:

– Формат V1 («Обычный экспорт») – используется для экспорта и импорта необходимых данных проекта. Поддерживается выбор типов точек данных, отдельных точек данных и конфигурационных элементов. Внутренняя структура соответствует типам точек данных и конфигурационным элементами. Информация, относящаяся к некоторому отдельному элементу точки данных, расположена в различных частях файла. Формат V1 поддерживает работу с многоязычными текстами (описания, тексты алармов).

– Формат V2 («Многоязычный экспорт») – обычно не требуется при разработке стандартных проектов.

– Формат V3 («Вывод PowerConfig (1 DP/строка, Формат3)») – используется только для работы с типами точек данных, для которых созданы мастер-точки данных. Как правило, для

каждого типа точек данных (класса устройств) создается отдельный файл. Информация, соответствующая каждой отдельной точке данных (каждому отдельному устройству), записывается в отдельную строку. Все логически связанные данные хранятся консолидированно в одной части файла. При использовании формата V3 сохраняются только те атрибуты, которые могут быть изменены на уровне точек данных. Формат может быть использован при массовом параметрировании.

– Формат V4 («Вывод PowerConfig (1 DPE/строка, Формат4)») – ориентирован на разработку систем для существующих установок или установок с большим количеством атрибутов точек данных. Данный формат схож с форматом V3. Отличие состоит в том, что информация о каждом отдельном элементе точки данных записывается в отдельную строку. Использование данного формата значительно облегчает ввод периферийных адресов устройств, описаний и т.п. во внешней программе для работы с электронными таблицами (с применением функции автозаполнения).

### 3.6 Создание списков точек данных

Редактор базы данных PARA позволяет создавать списки точек данных и шаблоны списков точек данных с целью их последующей обработки во внешних программах. Для этого используется пункт [Создать список DP для всех DPT (1DPE/строку)] контекстного меню, открываемого при нажатии ПКМ по требуемому типу точки данных.

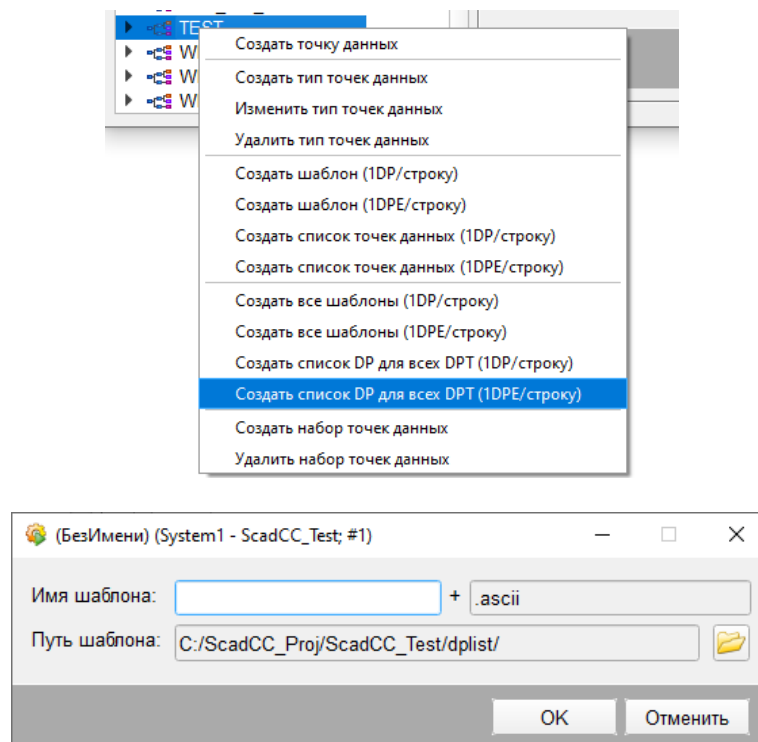


Рисунок 25 – Создание шаблонов и списков точек данных с целью их последующей обработки во внешних программах

### **3.7 Создание типов точек данных и точек данных с использованием сценариев**

В системе SCAD CC поддерживается создание типов точек данных и точек данных с использованием внутреннего языка сценариев Control. В языке Control имеются также функции для доступа к конфигурационным элементам и их настройки, для удаления или настройки интересующих точек данных и т.п.


Благодаря возможности конфигурирования системы во время работы проекта, для выполнения подобных операций остановка системы управления не требуется.

Основными функциями языка Control для работы с точками данных являются: "dpTypeCreate()", "dpCreate()", "dpCopy()", "dpDelete()", "dpSet()" и "dpSetDescription()".

## 4 Создание экранных форм

### 4.1 Графический редактор

В SCAD CC все графические экранные формы называются «панелями». Панели в системе SCAD CC могут использоваться для отображения обзорных экранных форм, экранных форм отдельных технологических процессов, информационных и диалоговых окон и т.п. Для создания панелей используется графический редактор GEDI. Для запуска графического редактора необходимо щелкнуть в консоли ПКМ по записи «Интерфейс пользователя -m gedi» и выбрать в открывшемся контекстном меню элемент [Запустить менеджер].

После открытия редактора нажмите на кнопку [Новая панель] . Некоторые настройки панелей целесообразно изменять только после открытия их в редакторе GEDI. Меню пиктограмм и другие элементы редактора GEDI могут быть перемещены при помощи мыши за пределы окна редактора.

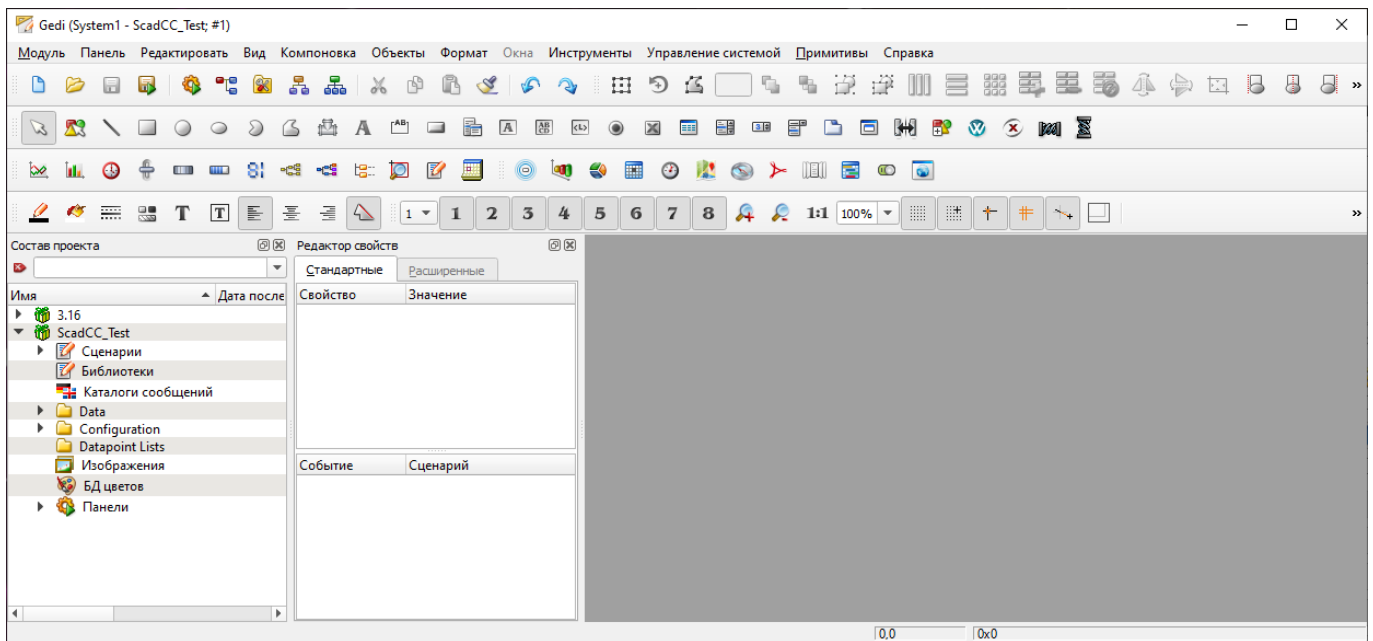


Рисунок 26 – Графический редактор GEDI

### 4.2 Меню пиктограмм

Окно редактора GEDI состоит из панели меню, панелей инструментов, состава проекта, редактора свойств, одного или нескольких окон панели, окна каталога, строки с заголовком и строки состояния.

На панелях инструментов функции сгруппированы по функциональному назначению.

Меню пиктограмм «Основные инструменты» выводится на экран по умолчанию. Данное меню пиктограмм можно скрыть нажатием ПКМ по меню пиктограмм и выбором опции «Основные инструменты».

Меню пиктограмм «Основные инструменты» (рисунок 27) содержит стандартные функции, которые также можно найти в меню «Панель» и «Редактировать».



Рисунок 27 – Меню пиктограмм «Основные инструменты»

Описание кнопок меню пиктограмм «Основные инструменты» представлено в таблице 1.

Таблица 1 – Описание кнопок меню пиктограмм «Основные инструменты»

Значок	Функция кнопок
	Открывает новую панель
	Открывает панель
	Сохраняет панель
	Сохраняет панель и запускает ее в модуле VISION
	Запускает модуль VISION
	Запускает модуль PARA
	Открывает панель «Управление системой»
	Открывает панель конфигурации иерархии
	Открывает панель редактора модели данных
	Вырезает, копирует, вставляет объекты
	Копирование формата
	Отменяет, повторяет действия

Инструменты создания базовых графических объектов включены в меню пиктограмм «Основные объекты». Меню пиктограмм объекта можно вывести на экран или скрыть нажатием ПКМ и выбором опции «Основные объекты» (рисунок 28).





















Рисунок 28 – Меню пиктограмм «Основные объекты»

В таблице 2 приведены список и описание кнопок меню пиктограмм «Основные объекты».

Таблица 2 – Описание кнопок меню пиктограмм «Основные объекты»

Значок	Функция кнопок
	Инструмент выбора
	Инструмент выбора объектов внутри ссылки
	Инструмент рисования линии
	Инструмент рисования прямоугольника

Значок	Функция кнопок
	Инструмент рисования круга
	Инструмент рисования эллипса
	Инструмент рисования сектора
	Инструмент рисования полигона
	Инструмент рисования трубопровода
	Инструмент создания текстового поля
	Инструмент рисования рамки
	Кнопка с меткой или изображением
	Каскадное меню
	Текстовое поле
	Текстовый редактор
	Объект Label
	Список кнопок-переключателей
	Список селективных кнопок
	Таблица
	Список выбора
	Счетчик
	Выпадающий список
	Вкладка
	Встроенный модуль
	Объект Splitter
	Вставляет панель-ссылку
	EWO (внешний виджет)
	Объект ActiveX
	Горизонтальный разделитель
	Вертикальный разделитель

В меню пиктограмм «Дополнительные объекты» включены инструменты для создания дополнительных графических объектов. Меню пиктограмм объекта можно вывести на экран или скрыть нажатием ПКМ и выбором опции «Дополнительные объекты» (рисунок 29).

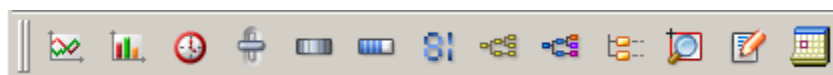















Рисунок 29 – Меню пиктограмм «Дополнительные объекты»

В таблице 3 приведен список и описание кнопок меню пиктограмм «Дополнительные объекты».

Таблица 3 – Описание кнопок меню пиктограмм «Дополнительные объекты»

Значок	Функции кнопок
	Тренд
	Столбчатый тренд
	Часы
	Ползунок
	Регулятор
	Виджет «Progress Bar»
	Часы в стиле ЖК-индикатора
	Дерево точек данных
	Тип точки данных
	Иерархический список
	Инструмент изменения масштаба
	Редактор сценариев
	Виджет «Календарь»







Меню пиктограмм «Инструменты форматирования» (рисунок 30) содержит инструменты форматирования объектов (форматирования символов текста, изменения цвета рисунка и цвета фона, ширины строки и т.д.).



Рисунок 30 – Меню пиктограмм «Инструменты форматирования»

В таблице 4 приведены список и описание кнопок меню пиктограмм «Инструменты форматирования».

Таблица 4 – Описание кнопок меню пиктограмм «Инструменты форматирования»

Значок	Функции кнопок
	Настройка цвета рисунка объекта
	Настройка цвета фона объекта
	Настройка ширины строки и стиля
	Настройка типа заливки
	Форматирование текста
	Текст с рамкой



Значок	Функции кнопок
	Выравнивание текста
	Замкнутый полигон

При помощи меню пиктограмм «Инструменты представления» (рисунок 31) можно настроить вид окна панели (задать различные слои, изменить язык, включить/отключить точки сетки).

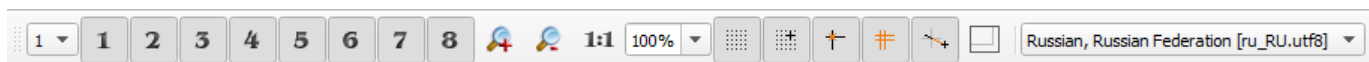


Рисунок 31 – Меню пиктограмм «Инструменты представления»

В таблице 5 приведен список и описание кнопок меню пиктограмм «Инструменты представления».

Таблица 5 – Описание кнопок меню пиктограмм «Инструменты представления»

Значок	Функции кнопок
	Настройка слоя графического объекта
	Увеличение или уменьшение масштаба
	Включение/отключение точек сетки
	Включение/отключение точек привязки
	Переключение привязки к узлам графического объекта
	Переключение привязки линии к узлам графического объекта
	Переключение привязки к продлению линии
	Границы стандартного размера для класса устройств
	Отображение активного языка предварительного просмотра панели

При помощи кнопок меню пиктограмм «Инструменты компоновки» (рисунок 32) можно запустить самые важные функции компоновки.













Рисунок 32 – Меню пиктограмм «Инструменты компоновки»

В таблице 6 приведены функции кнопок меню пиктограмм «Инструменты компоновки».

Таблица 6 – Функции кнопок меню пиктограмм «Инструменты компоновки»

Значок	Функция
	Переключение на режим перемещения/масштабирования
	Кнопка для поворота объекта при помощи манипулятора «мышь»

Значок	Функция
	Точки редактирования, контуры редактирования
	Поворот объектов на введенное числовое значение
	Перенос объекта на передний/задний план
	Построение группы
	Разбиение группы
	Кнопки управления компоновкой
	Горизонтальное отображение одного или нескольких объектов
	Вертикальное отображение одного или нескольких объектов
	Установка размера согласно классу устройств
	Выравнивание объектов













При помощи меню пиктограмм «EWO» (рисунок 33) на панель можно добавить внешние виджеты.



Рисунок 33 – Внешние виджеты

В таблице 7 приведены функции кнопок меню пиктограмм «EWO».


Таблица 7 – Функции кнопок меню пиктограмм «EWO»

Значок	Функция
	Добавляет внешний виджет AttentionEffect
	Добавляет внешний виджет BarChart3D
	Добавляет внешний виджет Chart
	Добавляет внешний виджет DateTimeEdit
	Добавляет внешний виджет DialGauge
	Добавляет внешний виджет GisGauge
	Добавляет внешний виджет Maps
	Добавляет внешний виджет Pdf
	Добавляет внешний виджет PictureFlow
	Добавляет внешний виджет Scheduler
	Добавляет внешний виджет ToggleSwitch
	Добавляет внешний виджет WebView


Панель меню GEDI содержит следующие пункты:

- Меню «Модуль»;
- Меню «Панель»;
- Меню «Редактировать»;
- Меню «Вид»;
- Меню «Компоновка»;
- Меню «Объект»;
- Меню «Формат»;
- Меню «Окна»;
- Меню «Инструменты»;
- Меню «Управление системой»;
- Меню «Примитивы»;
- Меню «Справка».

### 4.3 Основные операции в редакторе

Для добавления желаемого объекта на панель, например, линии, нажмите соответствующую кнопку в меню пиктограмм () , переместите курсор мыши в точку начала линии в рабочей области панели, нажмите и удерживайте нажатой левую кнопку мыши. Переместите курсор мыши в точку окончания линии и отпустите кнопку мыши для завершения создания линии. Линия будет добавлена на панель.

После создания объекта он автоматически остается выбранным. Для выбора другого объекта необходимо щелкнуть по нему ЛКМ. Выбранный объект обозначается обрамляющим прямоугольником с узловыми точками.

Для изменения размера объекта перетащите узловые точки в необходимом направлении при помощи мыши. При нахождении курсора мыши в пределах объекта SCAD CC автоматически активируется режим перемещения. При нахождении курсора мыши на границе объекта режим перемещения отображается значком курсора мыши  , после чего возможно перемещение объекта в пределах панели.


Для выбора нескольких объектов необходимо последовательно щелкнуть по ним левой кнопкой мыши, удерживая при этом в нажатом состоянии клавишу [Ctrl].

Во время выбора и рисования объектов, в строке состояния в нижней части окна GEDI выводятся координаты объекта и его размер.

Система координат редактора GEDI основана на стандарте Windows: точка отсчета (0,0) располагается в левом верхнем углу панели. Увеличение значения по оси X визуально соответствует движению вправо от нулевой точки. Увеличение значения по оси Y визуально

соответствует движению вниз от нулевой точки. Положительный угол вращения соответствует направлению вращения против часовой стрелки.

Для создания квадрата (прямоугольника с одинаковыми сторонами) при рисовании прямоугольника удерживайте клавишу [SHIFT].

В графическом редакторе возможен предварительный просмотр созданных панелей. Для отображения панели в режиме предварительного просмотра может быть использована кнопка  или элемент меню «Панель» → «Сохранить и запустить в модуле быстрого тестирования».

Внешний вид панели в режиме предварительного просмотра в графическом редакторе в значительной степени соответствует внешнему виду панели в модуле исполняемого пользовательского интерфейса (VISION). Однако в режиме предварительного просмотра некоторые операции могут выполняться дольше (сценарии, изменение цвета при наведении курсора и т.п.). Конструкции «картинка-в-картинке» отображаются корректно только в модуле VISION.

#### 4.4 Редактор свойств

Редактор свойств предназначен для просмотра и изменения свойств графических объектов. Редактор свойств может быть открыт при помощи меню «Вид» → «Редактор свойств». Редактор свойств содержит две вкладки, на которых выводится наиболее важная информация об объекте. Вторая вкладка «Расширенные [свойства]» активируется только для комплексных графических объектов, например, для ползунка, колеса прокрутки и т.д.

Редактор свойств может использоваться и для вывода информации о текущих значениях свойств, и для изменения свойств. Редактор свойств представляет собой дополнительный инструмент для управления свойствами графических объектов. При щелчке ЛКМ по области рисования панели, в редакторе свойств отображаются свойства панели. При помощи редактора свойств могут быть изменены, например, фон панели или размер (при помощи свойств "Размер x" и "Размер y").

Большое количество свойств графических объектов, заданных при помощи редактора свойств на этапе настройки, могут быть в дальнейшем изменены программным способом во время выполнения программы. Например, возможно изменение интересующего свойства объекта в зависимости от значения элемента точки данных.

Под «этапом настройки» выше подразумевается период разработки приложения. Выражение «во время выполнения программы» означает «в ходе работы запущенного проекта SCAD CC». Графический редактор используется для создания графических пользовательских интерфейсов, а модуль VISION используется для их отображения во время выполнения программы.

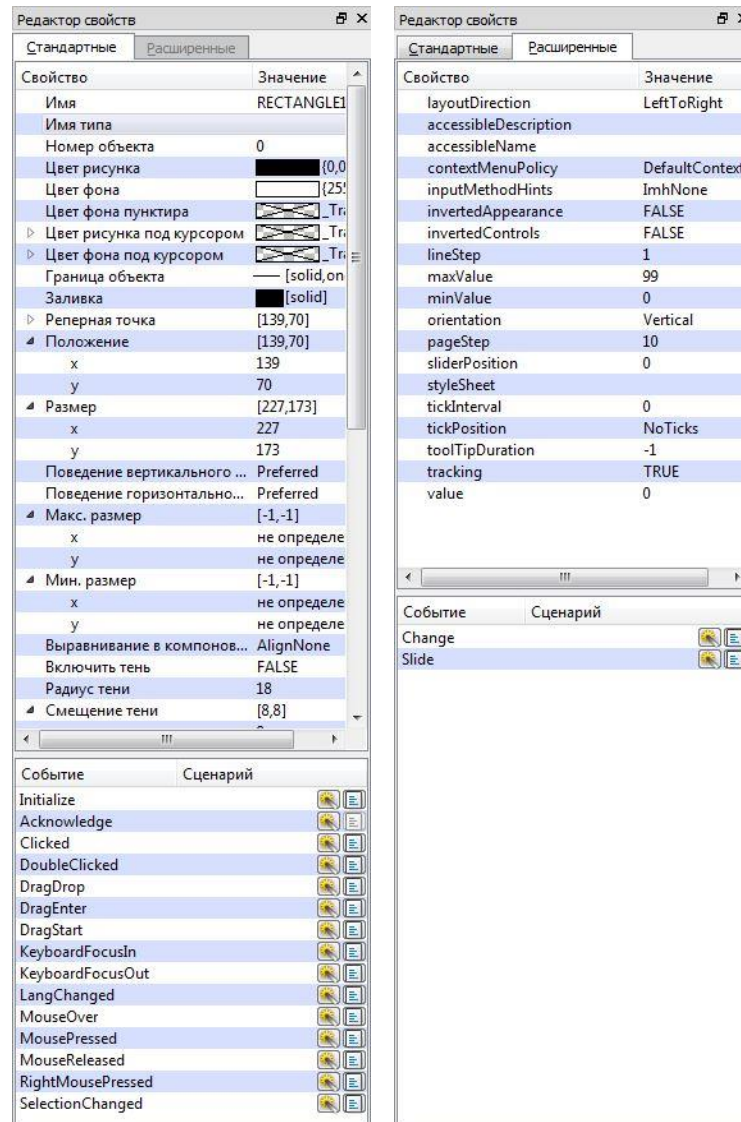


Рисунок 34 – Редактор свойств графических объектов

Вкладка «Событие» служит для задания действий, автоматически выполняемых при наступлении соответствующих событий во время работы программы. Примеры событий:

- открытие панели;
- щелчок левой кнопкой мыши по графическому объекту;
- подтверждение введенных данных путем нажатия клавиши [Ввод] ([Enter]);
- получение или потеря объектом фокуса и т.д.

При наступлении события выполнение соответствующего действия вызывается автоматически. Фактически действия определяются в сценариях.

#### 4.5 Модули, панели, дочерние панели

Системы управления реальными установками обычно включают в себя несколько десятков или сотен экранных форм, в связи с чем в подобных проектах требуется наличие соответствующих систем навигации и управления экранными формами (панелями).

SCAD CC поддерживает одновременную работу нескольких менеджеров интерфейса пользователя. При этом данные менеджеры интерфейса пользователя могут запускаться либо на одном компьютере (например, на сервере), либо на разных компьютерах. Как правило, для одного рабочего места настраивается один менеджер интерфейса пользователя.

Каждый менеджер интерфейса пользователя может использоваться для отображения одного или нескольких модулей визуализации. При этом для операционной системы каждый подобный модуль представляет собой отдельную программу.

В каждом модуле открывается так называемая «корневая панель». Корневая панель может представлять собой либо экранную форму процесса, либо панель с встроенными в нее другими модулями. При этом панели, открываемые в данных модулях, будут являться корневыми панелями.

Дополнительно имеется возможность открытия из модуля другого модуля в качестве независимого окна. При закрытии вызывающего модуля (из которого был запущен другой модуль) или при смене в нем конечной панели вызванный (новый) модуль не закрывается.

Подобный подход может применяться для отображения нескольких независимых друг от друга областей визуализации. Например, на компьютере с двумя мониторами на левом мониторе может постоянно отображаться экранная форма процесса, а на правом – панель алармов. При этом будет использоваться только один менеджер интерфейса пользователя.

Без применения дополнительных настроек в окне модуля по умолчанию всегда отображаются меню и меню пиктограмм. Однако из соображений безопасности в большинстве случаев данные элементы скрываются, а для навигации пользователю предоставляются только специально создаваемые на панели органы управления.

Из корневых панелей в качестве дочерних панелей могут открываться новые панели (окна). Дочерняя панель относится к той корневой панели, из которой она была открыта. При закрытии корневой панели соответствующие дочерние панели закрываются автоматически.

В зависимости от конкретной задачи модули и дочерние панели могут открываться различными способами.

Визуально разница между корневыми панелями и дочерними панелями незаметна, однако их поведение во время выполнения программы существенно различается.

## 5 Алармы

### 5.1 Настройка обработки алармов

Для элементов точек данных может быть настроена функциональность обработки алармов. При этом весь диапазон значений элемента точки данных разбивается на ряд непересекающихся и следующих непосредственно друг за другом аларм-диапазонов. В зависимости от нахождения значения элемента точки данных в пределах того или иного аларм-диапазона, квитирования ранее сформированных алармов и т.д. каждый из заданных аларм-диапазонов будет находиться в определенном состоянии. Двоичный элемент точки данных может принимать только два состояния. Диапазон аналоговых значений может быть разделен на практически неограниченное количество аларм-диапазонов. Соответственно, при настройке обработки алармов для двоичного элемента точки данных правила формирования алармов могут быть заданы только для одного из двух возможных состояний элемента точки данных. В случае диапазона аналоговых значений весь диапазон значений элемента точек данных, как правило, делится на 2, 3 или 5 аларм-диапазонов (максимум 255).

Настройка обработки алармов выполняется при помощи редактора PARA путем добавления к элементу точки данных конфигурационного элемента обработки алармов. Для этого:

- 1) щелкните по нужному элементу точки данных ПКМ и в открывшемся контекстном меню выберите «Добавить конф.элемент» → «Обработка алармов»;
- 2) откройте структуру конфигурационных элементов (двойным щелчком ЛКМ по элементу или нажатием на знак раскрытия перед именем элемента) и выберите конфигурационный элемент "\_alert\_hdl";
- 3) введите необходимые настройки;
- 4) нажмите кнопку [ОК] для сохранения настроек.

Настройки точки данных вступают в силу непосредственно после завершения описанных выше действий. Настройки применяются во время работы проекта и могут быть незамедлительно протестированы. Перезапуск системы не требуется.

При использовании мастер-точки данных конфигурационные элементы обработки алармов добавляются в виде шаблонных конфигурационных элементов непосредственно к интересующим элементам мастер-точки данных. Далее граничные значения и прочие настройки должны быть заданы/выполнены на уровне отдельных элементов точек данных. Соответствующий диалоговый интерфейс открывается при щелчке ПКМ по конфигурационному элементу обработки алармов настраиваемого элемента точки данных с последующим выбором элемента меню «Параметры шаблона конф.элемента».

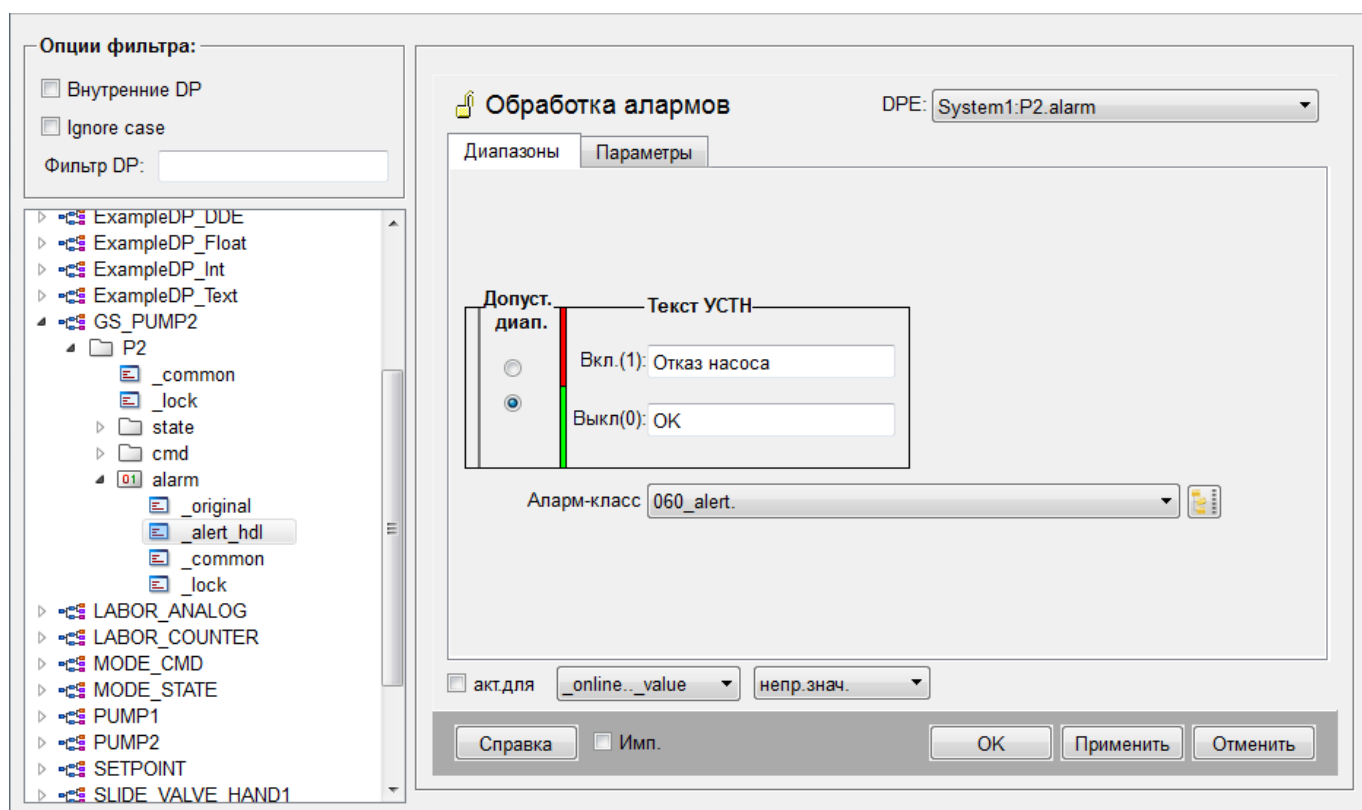


Рисунок 35 – Настройка обработки алармов для элемента точки данных

При работе без использования мастер-точек данных конфигурационные элементы обработки алармов ("\_alert\_hdl") добавляются непосредственно к интересующим элементам точек данных.

При настройке обработки алармов для аналоговых значений для каждого из аларм-диапазонов может быть активирован гистерезис. Гистерезис позволяет избежать формирования многочисленных алармов при незначительных по амплитуде, но частых колебаниях значения вокруг граничного значения аларм-диапазона. Для каждого из аларм-диапазонов может быть настроен ряд параметров. К таким параметрам, например, относятся:

- Приоритет аларма (приоритет аларм-диапазона);
- Модель состояний алармирования;
- Полномочия на квитирование алармов;
- Цвет индикации аларма;
- Необходимость формирования и хранения истории алармов;
- Действия по аларму и т.д.

Во избежание повторной индивидуальной настройки данных параметров для каждого из аларм-диапазонов, в системе SCAD CC используются так называемые аларм-классы, включающие в себя наборы предварительно выполненных настроек. При этом для применения к аларм-диапазону интересующего набора предварительно выполненных настроек достаточно указать соответствующий аларм-класс. На рисунке ниже поля для задания аларм-классов для аларм-



диапазонов находятся справа от полей для ввода граничных значений. Средняя строка соответствует «диапазону допустимых значений», для которого настройка правил формирования алармов и, соответственно, задание аларм-класса не требуются.

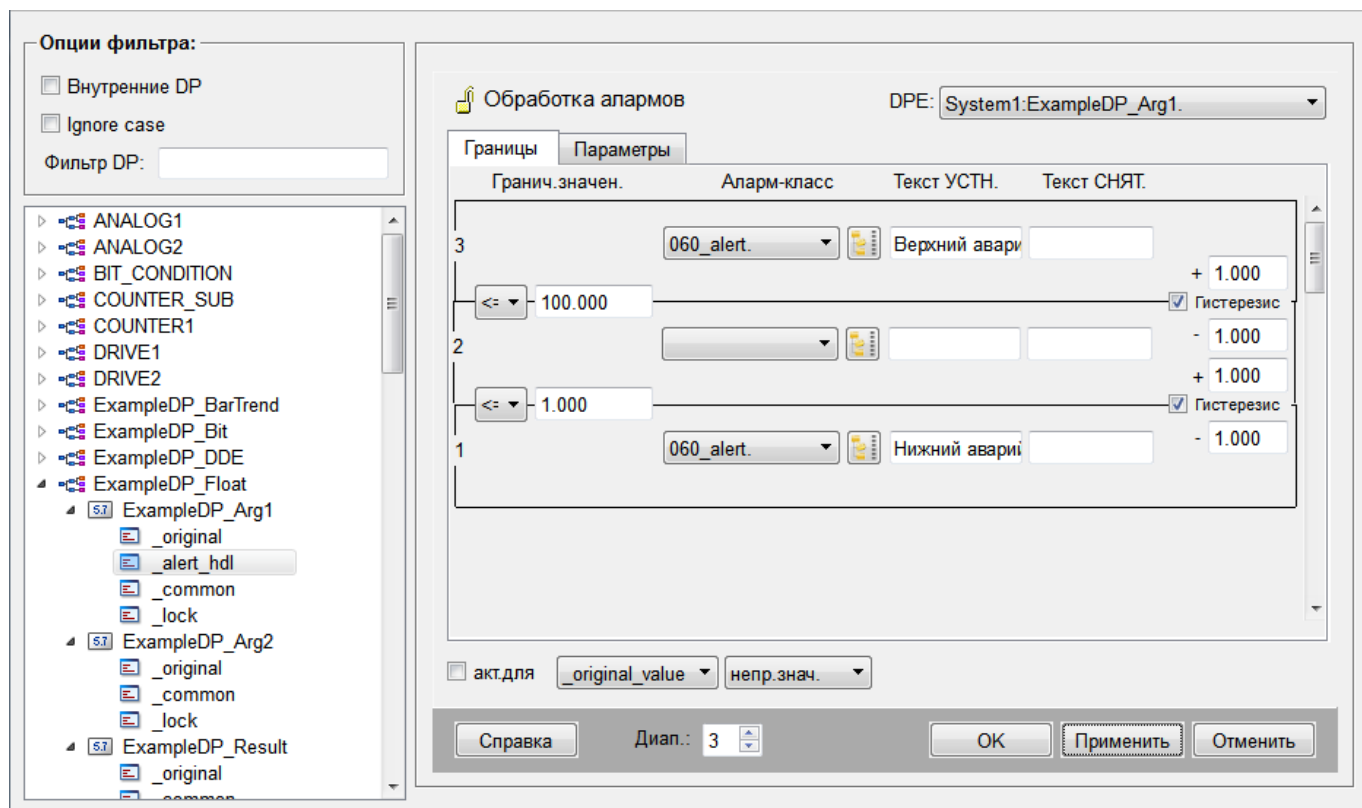


Рисунок 36 – Настройка обработки алармов с тремя диапазонами

В данном примере используется аларм-класс "060\_alert" с приоритетом 60 «тревога» (alert). Аларм-классы настраиваются в системе SCAD CC с использованием конфигурационных элементов "\_alert\_class". Все существующие аларм-классы настроены на базе внутренних точек данных типа "\_AlertClass" и могут быть просмотрены в редакторе PARA. Для просмотра существующих аларм-классов в редакторе PARA необходимо установить флажок «Внутренние DP» и в древовидной структуре точек данных перейти к типу "\_AlertClass".

## 5.2 Состояния аларм-диапазона и квитирование

Алармы используются для информирования пользователя о нештатных или опасных состояниях установки. Как правило, пользователь должен подтверждать факт приема аларма определенного приоритета. Подобное подтверждение получения информации (аларма) именуется «квитированием». Информация о квитировании, в общем случае, сохраняется совместно с идентификатором пользователя, выполнившего квитирование.

Состояние аларм-диапазона может изменяться во времени и зависит от нахождения значения элемента точки данных в пределах аларм-диапазона и наличия неквитированных алармов.

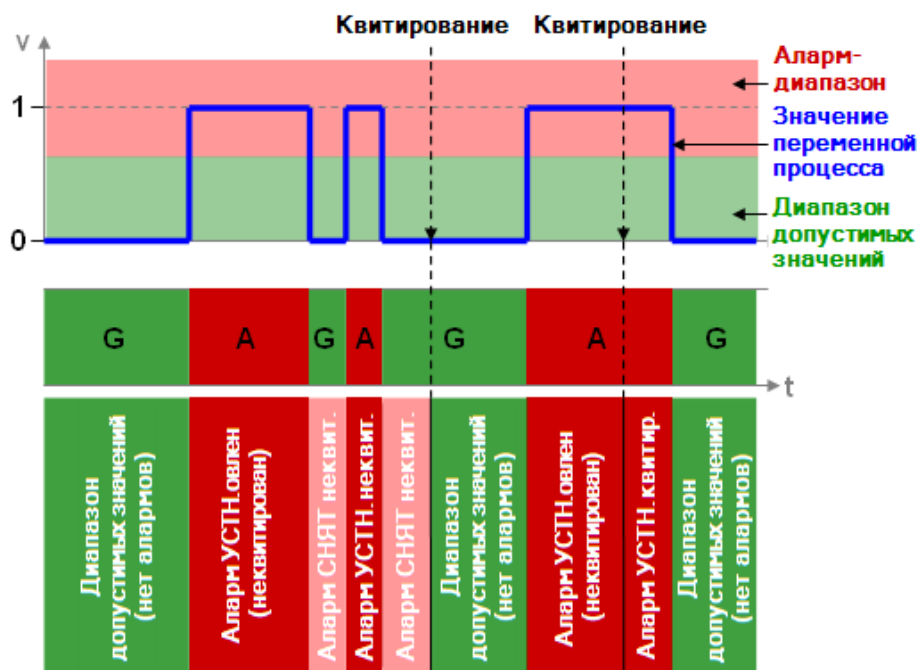


Рисунок 37 – Состояния аларм-диапазона в зависимости от значения и квтирования алармов

Необходимость квтирования определяется требованиями по безопасности. Обязательное квтирование может быть настроено для событий входа значения в аларм-диапазон, выхода значения из аларм-диапазона, для обоих данных событий или ни для одного из них.

Как правило, возможно одновременное квтирование всех алармов, которые в настоящий момент отображаются на текущей экранной форме процесса. Подобное квтирование в системе SCAD CC именуется «Общим квтированием» или «Квтированием набора алармов». Для некоторых алармов может быть настроена необходимость их отдельного квтирования. Подобные настройки выполняются на уровне аларм-класса.

### 5.3 Цвета алармов, мигание

SCAD CC поддерживает сопоставление различных цветов различным состояниям аларм-диапазона с возможностью соответствующего последующего окрашивания интересующих графических элементов экранных форм. При реализации цветовой индикации поддерживаются не только статичные цвета, но и мигание, то есть чередование двух или более цветов.

Согласно международным директивам (VDI 3699, DIN 19235, ...) приоритет аларма отображается цветом, а частота мигания используется для отображения текущего состояния аларм-диапазона.

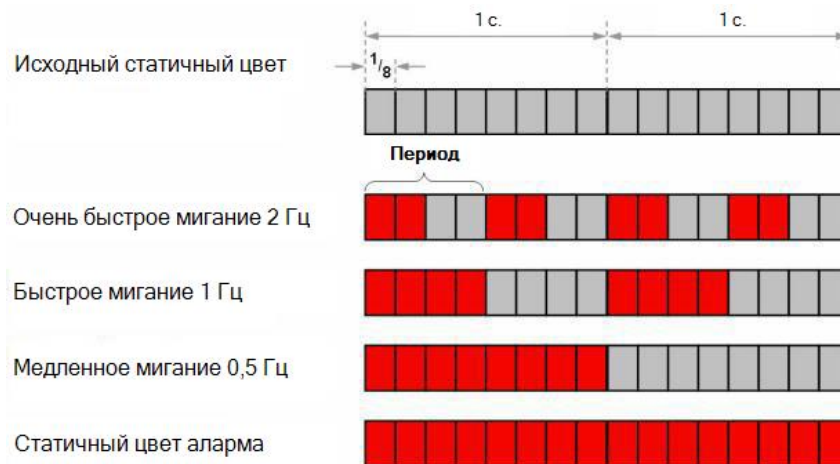


Рисунок 38 – Использование мигания для отображения текущего состояния аларм-диапазона

Настройка цветов выполняется на уровне аларм-класса. Система обработки алармов SCAD СС позволяет оценивать цвета, соответствующие состояниям аларм-диапазонов, и отображать данные цвета на экране.

Панель выбора цвета может быть вызвана не только из мастера упрощенного параметрирования, но и из редактора сценариев. В том случае, если состояние аларм-диапазона не подразумевает квитирования, или если необходимость вывода данного состояния отсутствует, объект отображается в исходном цвете, то есть в том цвете, который использовался при создании объекта. Цвета для цветовой анимации (мигания) могут быть выбраны в панели выбора цвета по их названиям, а также заданы вручную.

Алармы на экранной форме процесса могут выводиться различными способами. Аларм может отображаться, например, путем изменения цвета фона или цвета рисунка символа устройства. При этом цвет фона или цвет рисунка объекта будет соответствовать цвету аларма. При отсутствии аларма объект будет отображаться в исходном цвете, т.е. в том, который использовался при создании объекта. Подобная функциональность может быть настроена при помощи мастера свойств.

## 5.4 Панель алармов

SCAD СС включает в себя высокофункциональную и гибко настраиваемую панель алармов, именуемую также «Панелью алармов и событий». Данная панель может быть открыта из панели управления системой, раздел «Диагностика».

Ниже представлена панель алармов, вызываемая непосредственно из интерфейса пользователя. Интересующий аларм может быть квитирован щелчком ЛКМ по трем восклицательным знакам в ячейке в столбце «Квитирование». Аларм также может быть квитирован при помощи контекстного меню, вызываемого щелчком ПКМ по соответствующей строке.

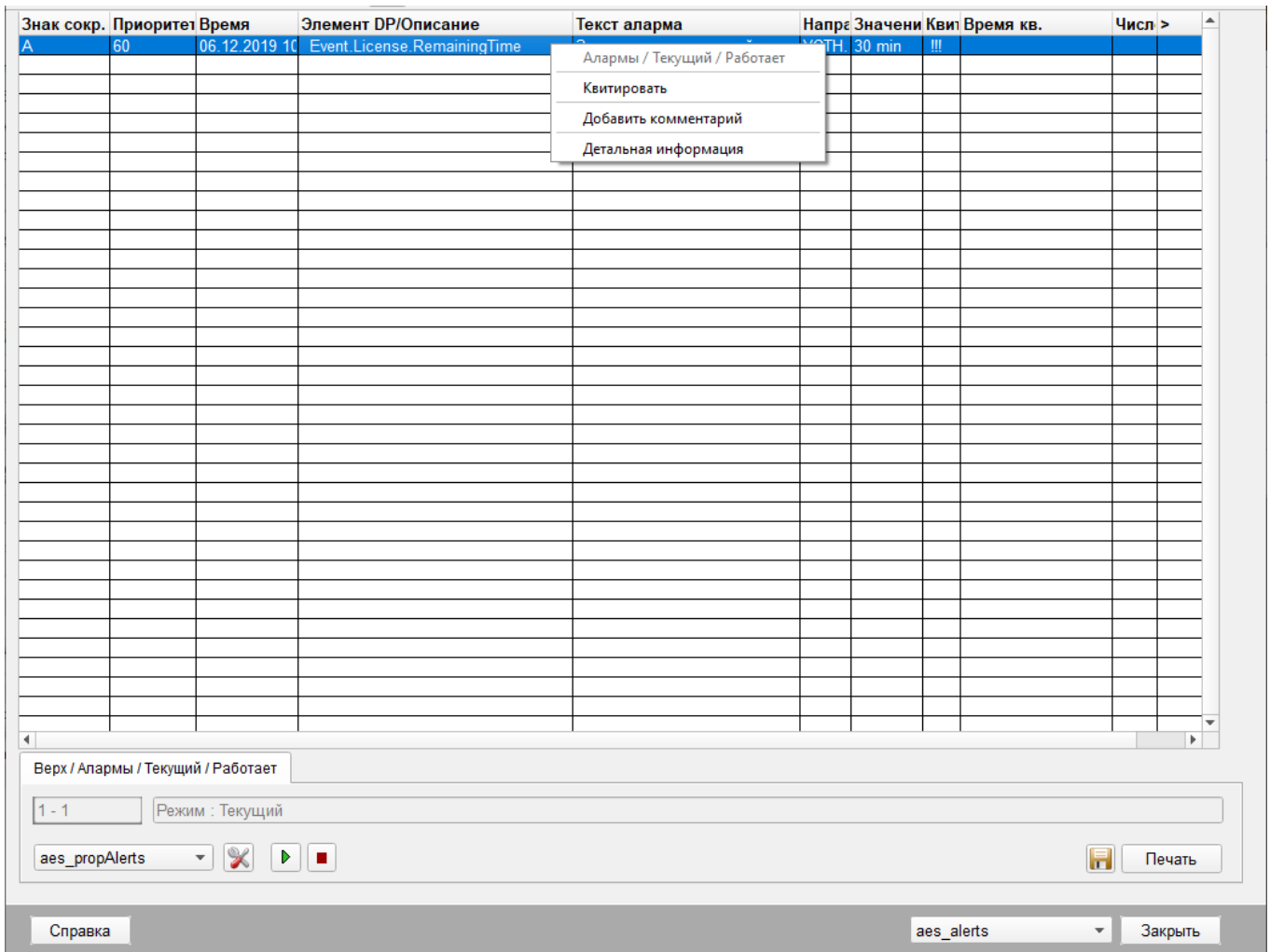


Рисунок 39 – Панель алармов SCAD CC

После открытия панели алармов, для ее активации необходимо нажать кнопку с зеленым треугольником слева внизу панели. После активации панели в ней будут отображаться текущие алармы. Подобная последовательность обеспечивает возможность предварительного выбора ранее созданной конфигурации (слева внизу) без необходимости отдельного запуска запроса, выполняемого в фоновом режиме и формирующего набор данных для отображения в панели алармов.

Первое открытие панели алармов или строки алармов может выполняться с задержкой в несколько секунд, поскольку сперва в фоновом режиме системой должны быть выполнены некоторые внутренние настройки.

По умолчанию панель алармов всегда открывается на переднем плане. То есть она располагается поверх всех открытых окон, при этом даже новые окна открываются за панелью алармов. Данное поведение окна панели алармов может быть изменено в ее настройках. Доступ к настройкам осуществляется путем нажатия кнопки [Настройка] на панели алармов.

Панель алармов может также использоваться в конфигурации одновременного отображения не только алармов, но и событий. И панель алармов, и панель алармов и событий

являются гибко настраиваемыми. Возможна настройка цветов, стилей, компоновки и ширины столбцов. Также имеется возможность добавления новых столбцов для вывода каких-либо дополнительных данных.

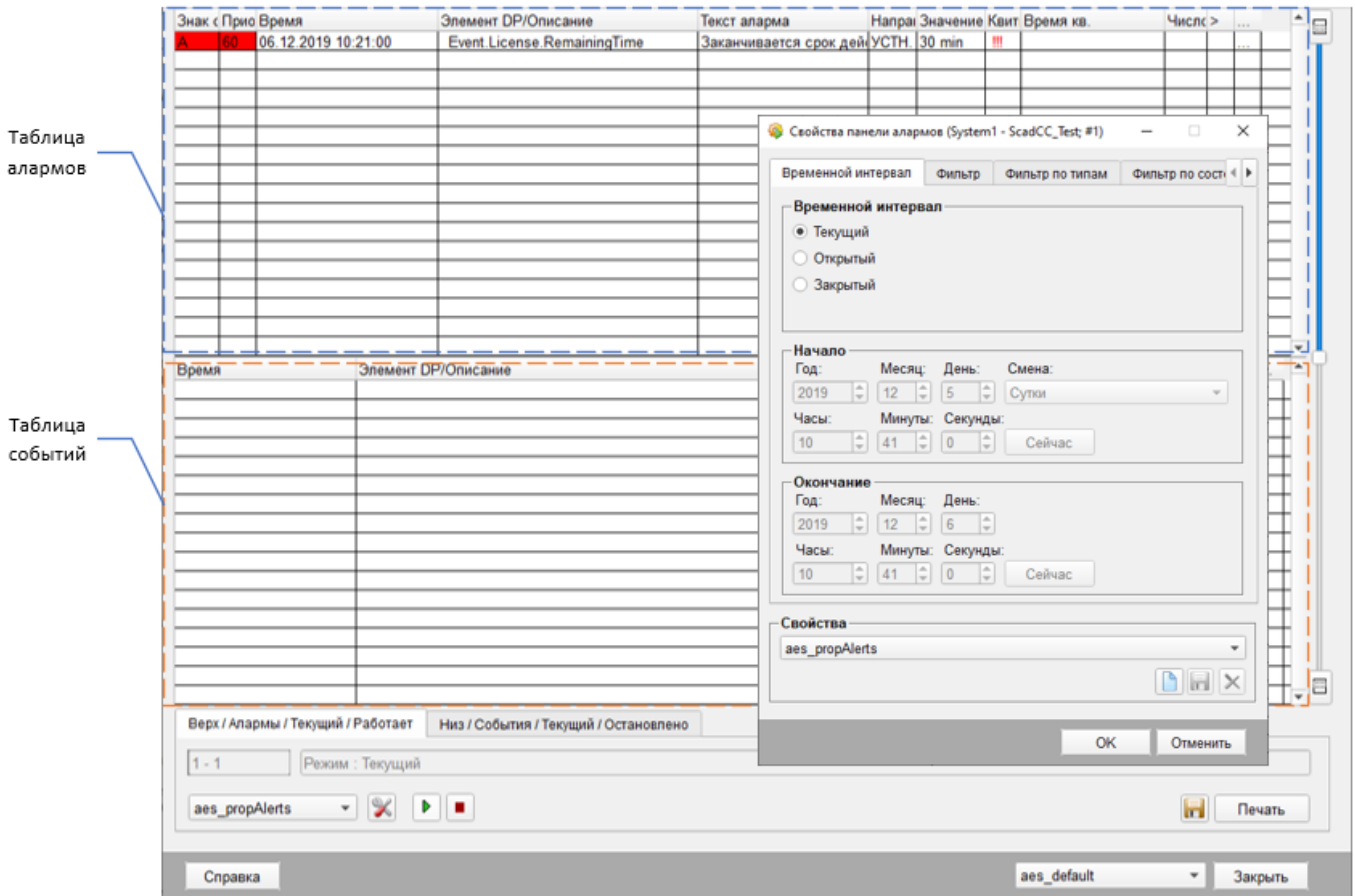


Рисунок 40 – Панель алармов и событий SCAD CC

## 6 Архивирование

### 6.1 Активация ведения истории значений

Система SCAD CC автоматически сохраняет последние значения процесса, что позволяет восстанавливать последнее состояние системы при перезапуске проекта.

SCAD CC поддерживает ведение истории алармов и значений. Архивирование настраивается на уровне отдельных элементов точек данных при помощи конфигурационных элементов или шаблонных конфигурационных элементов.

Для настройки архивирования:

- 1) откройте редактор базы данных PARA;
- 2) выберите тип точки данных, а затем мастер-точку данных этого типа;
- 3) щелкните ПКМ по элементу, для которого требуется архивирование, и выберите в открывшемся контекстном меню элемент [Добавить шабл.конф.элемент];
- 4) в открывшемся окне установите флажок [Архивирование];
- 5) в выпадающем списке выберите элемент [Фиксир.], что означает, что выполненные настройки будут применены ко всем аналогичным элементам точек данных этого типа без возможности их корректировки на уровне элементов точек данных;
- 6) для подтверждения выполненных настроек нажмите кнопку [ОК];
- 7) далее необходимо выбрать архив, в котором будут сохраняться исторические значения. Установите флажок [Активно] и выберите в выпадающем списке архив "01) 5-ти минутный архив". Имя архива говорит о том, что он предназначен для хранения значений, обновляемых примерно 1 раз в течение каждых 5 минут. Однако при быстрых изменениях значения, например 1 раз в секунду, все изменения также будут сохраняться в архиве. Каждое новое значение сохраняется исключительно при каждом изменении значения (при деактивированном сглаживании), при этом при быстрых изменениях значения все изменения также архивируются;
- 8) для подтверждения выполненных настроек нажмите кнопку [ОК]. Настройки архивирования будут автоматически применены к точкам данных выбранного типа.

В системе SCAD CC возможна одновременная работа нескольких менеджеров архивов, и, соответственно, возможна одновременная запись значений в нескольких автономных архивах.

В том случае, если мастер-точка данных не используется, для активации архивирования значений элементов точек данных конфигурационные элементы архивирования необходимо добавлять непосредственно к требуемым элементам точек данных. Для добавления конфигурационного элемента архивирования необходимо щелкнуть по нужному элементу точки данных ПКМ и выбрать в открывшемся контекстном меню [Добавить конф.элемент] → [Архивирование]. Далее необходимо выбрать ЛКМ созданный конфигурационный элемент

"\_archive" и задать требуемые настройки. Настройки будут относиться только к данному элементу, а не ко всем аналогичным элементам всех точек данных того же типа.

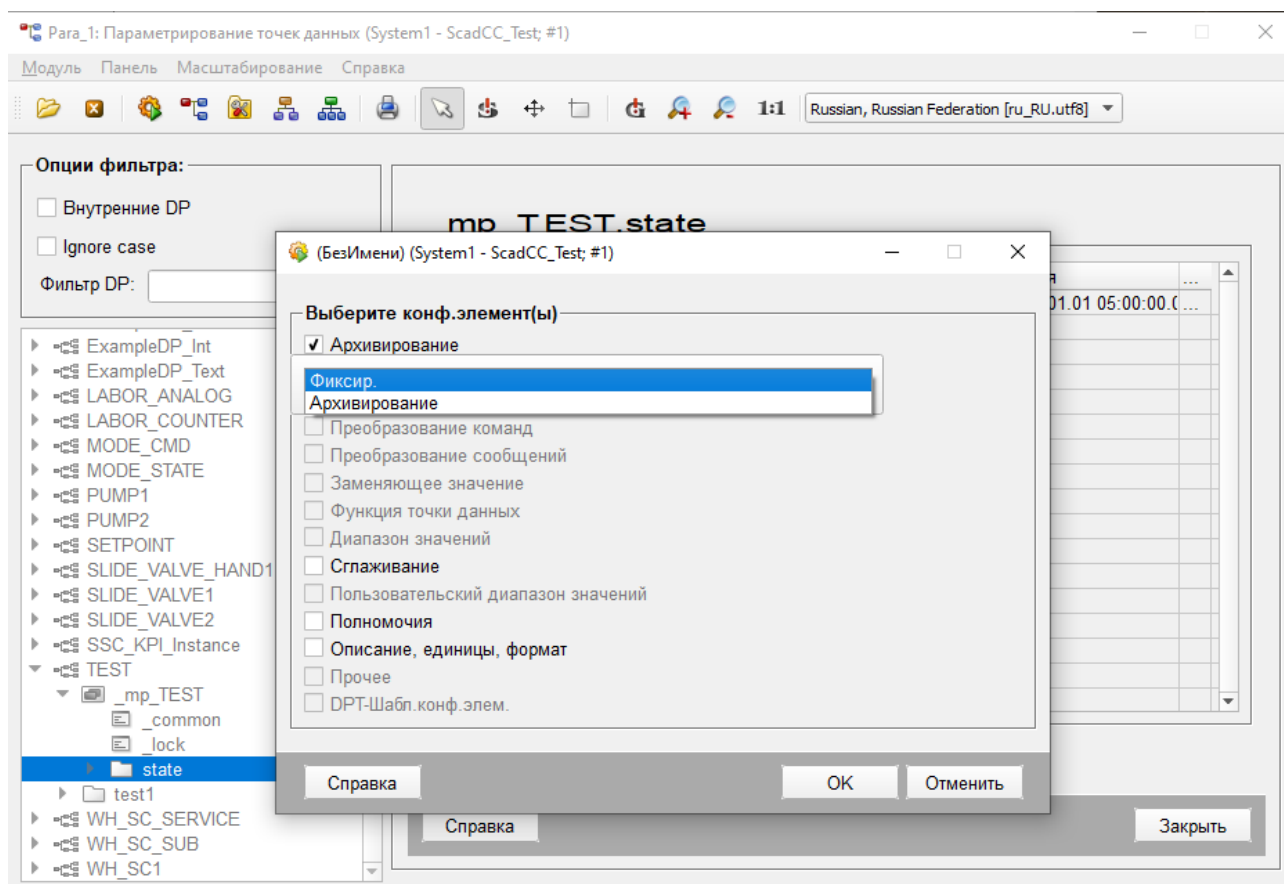


Рисунок 41 – Добавление к мастер-точке данных шаблонного конфигурационного элемента архивирования

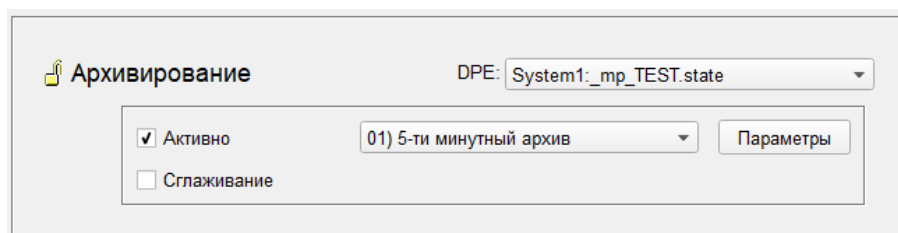


Рисунок 42 – Выбор целевого архива

В системе SCAD CC возможна одновременная работа нескольких менеджеров архивов, и, соответственно, возможна одновременная запись значений в нескольких автономных архивах.

В том случае, если мастер-точка данных не используется, для активации архивирования значений элементов точек данных конфигурационные элементы архивирования необходимо добавлять непосредственно к требуемым элементам точек данных. Для добавления конфигурационного элемента архивирования необходимо щелкнуть по нужному элементу точки данных ПКМ и выбрать в открывшемся контекстном меню [Добавить конф.элемент] > [Архивирование]. Далее необходимо выбрать ЛКМ созданный конфигурационный элемент

"\_archive" и задать требуемые настройки. Настройки будут относиться только к данному элементу, а не ко всем аналогичным элементам всех точек данных того же типа.

## 6.2 Отображение исторических данных с помощью трендов

SCAD CC обладает набором возможностей для отображения и обработки исторических данных. Одним из наиболее важных инструментов является график зависимости значений от времени («тренд»).

В системе SCAD CC существует уже готовое решение, позволяющее графически отображать зависимости значений от времени и при этом изменять интересующие настройки непосредственно в интерфейсе пользователя во время выполнения программы. Данное решение именуется «Конфигурируемым трендом».

Во время выполнения программы в конфигурируемом тренде могут быть изменены, например, источники отображаемых данных, внешний вид кривых и другие настройки. Настройки могут быть сохранены в виде конфигураций и впоследствии загружены также и во время выполнения программы.

Конфигурируемый тренд может быть открыт из панели управления системой: «Управление системой» → «Отчеты» → «Тренды». Для запуска данного инструмента, например, из экранной форме процесса, может быть создана и настроена соответствующая кнопка.

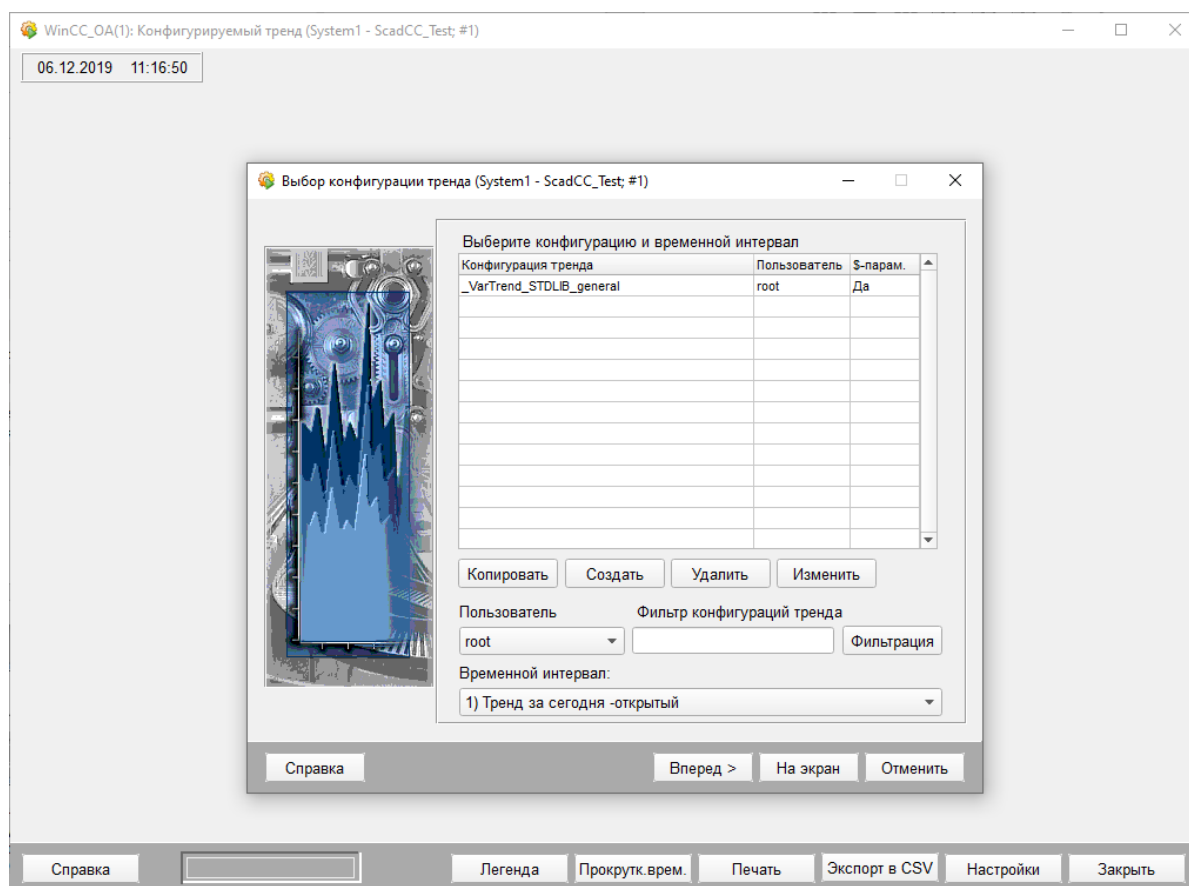


Рисунок 43 – Конфигурируемый тренд



## 7 Создание сценариев

### 7.1 Общая информация

SCAD CC включает в себя мощный язык сценариев. Данный язык может быть использован для реализации пользовательской логики в диалоговых окнах, примитивах (шаблонных символах устройств), для выполнения непрерывных расчетов, формирования отчетности и т.д. В зависимости от реализуемой задачи, сценарии могут запускаться как в пределах интерфейсов пользователя (при наступлении некоторого события, например, при щелчке кнопкой мыши по графическому объекту и т.п.), так и в рамках специально выделенных для этих целей менеджеров сценариев. Кроме этого, с помощью языка Control могут быть определены арифметические функции для точек данных. При этом данные функции будут выполняться в менеджере событий.

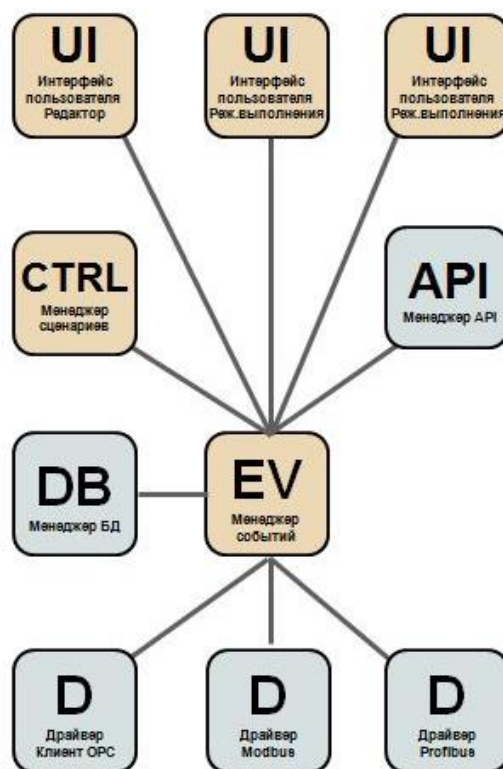


Рисунок 44 – Использование сценариев на языке Control в менеджерах SCAD CC

Control представляет собой процедурно-ориентированный язык высокого уровня с поддержкой всех типовых управляющих структур. Синтаксис языка близок к стандарту ANSI-C. Некоторые упрощения направлены на повышение удобства использования. В состав SCAD CC включен ряд функций, специально разработанных для решения задач в области автоматизации:

- Доступ к элементам точек данных (к переменным образа текущего процесса);
- Доступ к истории значений и алармов;
- Доступ к конфигурационным элементам элементов точек данных;
- Доступ к свойствам и методам графических объектов;

– Доступ на уровне операционной системы к файлам, последовательному интерфейсу, подключениям TCP, внешним базам данных, электронной почте, SMS и т.д.

Код обрабатывается интерпретатором, в связи с чем дополнительные компилирование или сборка со стороны пользователя не требуются. Сценарии, как правило, выполняются по событиям, но также поддерживается их циклическое выполнение или выполнение по заданному временному графику.

Сценарии на языке Control, используемые для реализации логики в диалоговых окнах или примитивах (шаблонных символах устройств) обычно запускаются при наступлении каких-либо событий (например, при щелчке кнопкой мыши по объекту и т.п.).

Из сценария Control, запускаемого при наступлении интересующего события объекта, возможен доступ не только к свойствам и методам данного объектов, но и доступ к любым другим свойствам и методам графических объектов, находящимся на любых других экранных формах проекта.

## 7.2 Редактор сценариев Control

Для создания сценария необходимо запустить графический редактор GEDI. В меню графического редактора следует открыть вкладку «Редактировать» и выбрать пункт [Новый сценарий CTRL...].

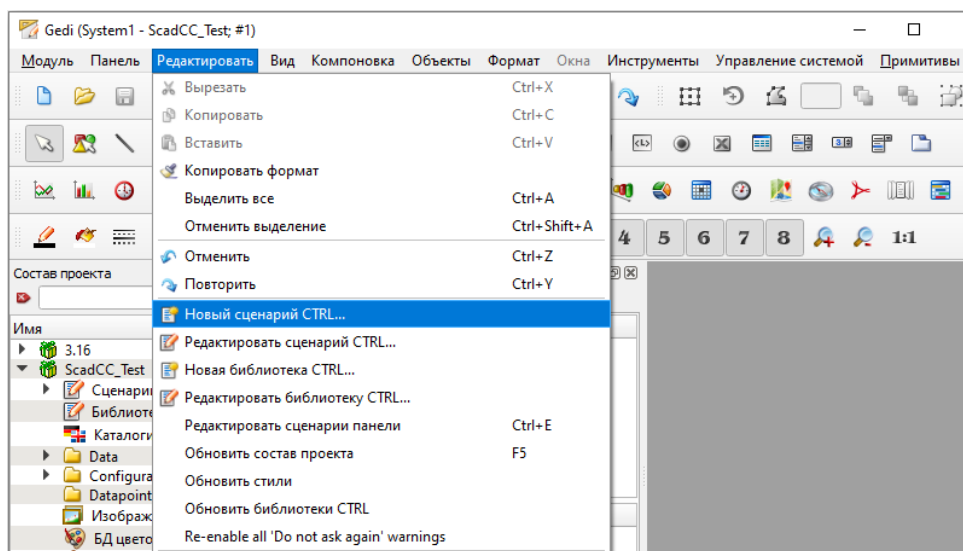


Рисунок 45 – Открытие редактора сценариев Control

Редактор сценариев снабжен набором помощников при вводе. Данные помощники могут быть открыты из меню «Инструменты». Помощники при вводе позволяют добавлять в текст редактируемого сценария следующие фрагменты:

Пункт меню «Инструменты»	Добавляемый фрагмент
Выбор панели	Имя файла панели (имя файла экранной формы, диалогового окна и

Пункт меню «Инструменты»	Добавляемый фрагмент
	т.п.)
Выбор цвета	Строка управления цветом (цвет может быть указан либо в кодировке RGB, либо с использованием названия цвета)
Выбор шрифта	Строка управления шрифтом
Выбор файла	Строка управления файлом
Выбор типа заливки	Строка управления заливкой (тип заливки: контур, сплошная, рисунок, штриховка, градиент)
Выбор типа линии	Строка управления типом линии (ширина линии, стили, соединения, окончания)
Выбор формата	Строка управления форматом
Выбор точки данных	Идентификатор точки данных, элемента точки данных, конфигурационного элемента, атрибута

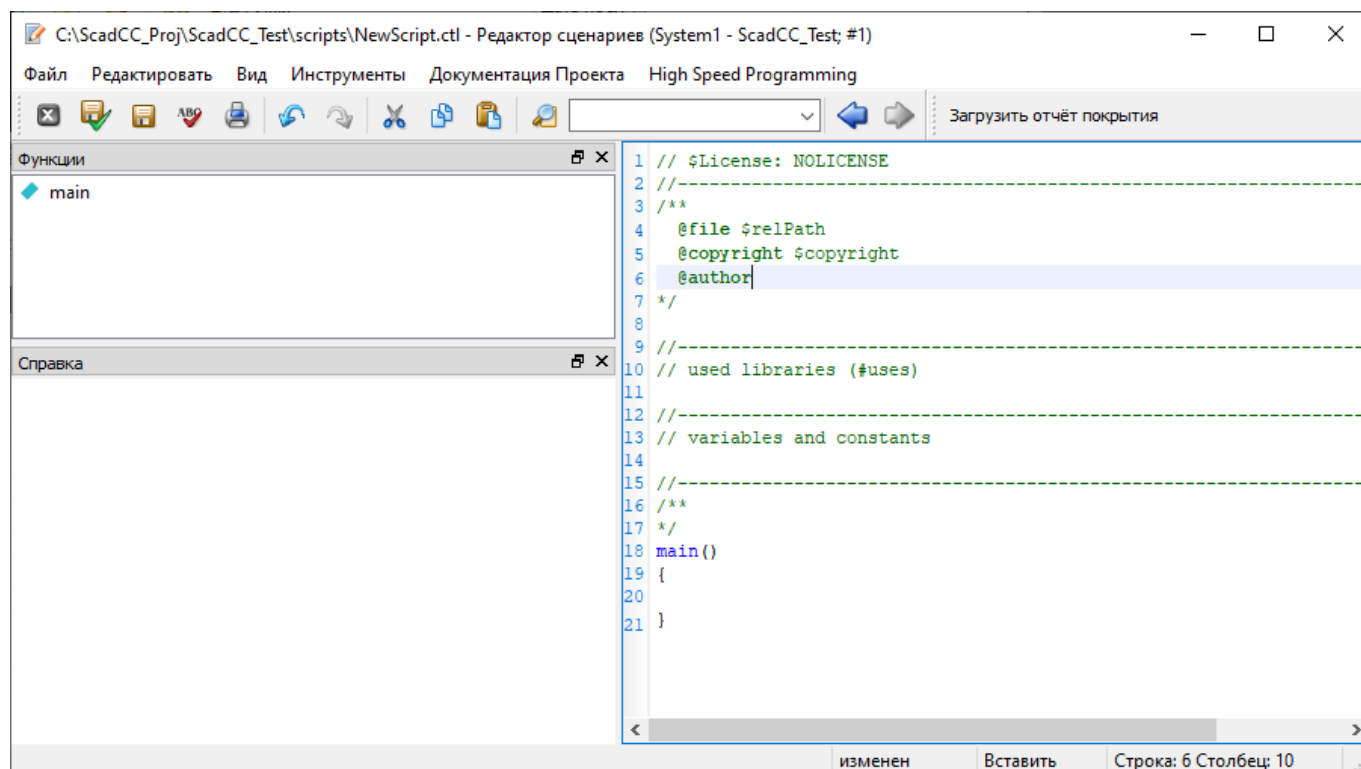


Рисунок 46 – Редактор сценариев Control

### 7.3 Структура сценариев и синтаксис

Каждый сценарий на языке Control в обязательном порядке содержит функцию "main()", которая не возвращает значений (void) и, в большинстве случаев, не имеет параметров. Собственно код заключается в фигурные скобки, которые также используются для выделения частей кода.

```
main()
{
```

```
// Комментарий: далее будет следовать код программы
```

```
/* Многострочный  
комментарий */  
}
```

Начало однострочного комментария обозначается комбинацией символов `//`, а окончание определяется концом строки. Начало многострочного комментария обозначается комбинацией символов `/*`, а конец `*/`.

Каждое утверждение (statement) должно заканчиваться точкой с запятой `;`. Утверждения могут находиться как в одной строке, так и в отдельных строках, что не влияет на выполнение кода. Тем не менее, с целью повышения наглядности кода рекомендуется его визуальное структурирование, в том числе с использованием отступов.

## 7.4 Переменные, константы, типы данных

В начале каждой программы должны быть объявлены локальные переменные и константы. В сценарии они представляют собой локальные носители информации, которые существуют независимо от элементов точек данных и хранятся в оперативной памяти отдельно. Для каждой переменной должен быть определен тип данных.

```
main()  
{  
int a; // Переменная 'a' типа 'integer' (int, целое)  
float b,c,d,e; // Набор переменных типа 'float'  
bool switchOn; // Переменная типа 'bool'  
const float pi = 3.1415 // Объявление константы типа 'float'  
// Далее может следовать часть кода  
// ...  
b = c + d - e; // Арифметическое выражение и присвоение значения переменной  
float f; // Объявление переменной внутри кода  
f = 42.42;  
b = b + f;  
// Далее может следовать часть кода  
}
```

Переменные могут объявляться в любой части кода программы. Ключевое слово `"const"` позволяет обеспечить неизменность значения переменной. Пример объявления переменной с

неизменным значением (объявление именованной константы): `const float pi = 3.1415` (объявление с инициализацией).

В рамках одного утверждения могут быть объявлены несколько переменных одного типа. Объявляемые таким образом переменные должны располагаться после ключевого слова, определяющего тип данных, и разделяться запятыми " , ". После объявления переменных (-ой), а также в конце каждого утверждения (statement) ставится точка с запятой " ; ".

Полный набор поддерживаемых типов данных включает в себя типы данных элементов точек данных, а также типы данных, допускающие изменение типа во время выполнения программы, например, "anytype" или "mixed". Также поддерживаются ассоциативные массивы.

Помимо локальных переменных, существующих в рамках отдельного сценария, в языке Control поддерживается возможность использования переменных с различными областями видимости:

- Переменные с областью видимости в пределах сценария (объявление до утверждения "main()" текущей программы);
- Переменные с областью видимости в пределах панели (объявление в сценарии события "ScoreLib" панели до выполнения каких-либо функций);
- Переменные с областью видимости в пределах менеджера (объявление при помощи функции "addGlobal" языка Control или объявление глобальных переменных в библиотеке, подключаемой непосредственно при запуске менеджера, до выполнения каких-либо функций).

## 7.5 Управляющие структуры

В языке Control доступны все стандартные управляющие структуры, соответствующие синтаксису ANSI-C.

```
if (<проверочное_выражение_A>
{
    <утверждение(я)_A >
    // Код, исполняемый в том случае, если значение проверочного выражения А равно "истина"
}
else if (<проверочное_выражение_B>)
{
    <утверждение(я)_B >
    // Код, исполняемый в том случае, если значение проверочного выражения В равно "истина"
}
```

```
else
```

```
  <утверждение(я)_C>
```

```
  // Утверждения_C выполняются только тогда,
```

```
  // когда ни значение проверочного выражения А, ни значение проверочного выражения В не равны
```

```
"TRUE" ("истина").
```

```
  // Одиночные утверждения могут не заключаться в фигурные скобки
```

Все проверочные выражения в конструкции "if/else" должны возвращать двоичные значения (т.е. "TRUE" или "FALSE").

```
switch(<проверочное_выражение>
```

```
{
```

```
  case(<выражение1>): {
```

```
    <утверждение(я)_1>
```

```
    // ... Код
```

```
  break;
```

```
}
```

```
  case(<выражение2>): {
```

```
    <утверждение(я)_2>
```

```
    // ... Код
```

```
  break;
```

```
}
```

```
//...
```

```
  case(<выражениеп>): {
```

```
    <утверждение(я)_п>
```

```
    // ... Код
```

```
  break;
```

```
}
```

```
  default: {
```

```
    // ... Код
```

```
  break;
```

```
}
```

```
}
```

Проверочное выражение в конструкции "switch/case" должно возвращать значение, которое может быть сравнено с выражениями после меток "case".

```
for (i = 1; i <= 15; i++) // i++ означает i = i + 1
{
    // Код, исполняемый циклично
}
```

Последовательность из нескольких утверждений в теле цикла "for" выполняется установленное количество раз. Как правило, указываются начальное значение, конечное значение и шаг счетчика цикла. В приведенном выше примере в качестве счетчика цикла используется переменная "i". Цикл выполняется ровно 15 раз, при этом начальным значением переменной "i" является 1, а непосредственно после каждого выполнения цикла значение переменной "i" увеличивается на единицу.

```
while (<проверочное_выражение>
{
    // Непрерывно исполняемый программный код
}

do
{
    // Непрерывно исполняемый программный код
} while (<проверочное_выражение>)
```

Конструкция "while" позволяет повторять некоторый набор утверждений до тех пор, пока значение проверочного выражения равно "TRUE" ("истина"). В том случае, если на момент начала обработки конструкции значение проверочного выражения не равно "истине", тело цикла не будет выполнено ни разу.

Управляющая конструкция "do-while" аналогична циклу "while". В отличие от цикла "while" утверждения в теле цикла "do-while" выполняются хотя бы 1 раз, поскольку значение проверочного выражения оценивается после тела цикла. Оператор "break" используется в управляющих конструкциях "switch" и циклах "for", "while" и "do-while" для принудительного выхода из управляющей конструкции.

## 7.6 Программы, функции, библиотеки

В языке Control каждая программа представляет собой автономную последовательность исполняемых команд. Синтаксически определение программы аналогично определению функции, однако в определении программы используется фиксированное имя "main()", при этом программа не возвращает значений (void) и, как правило, не использует параметры.

Язык Control, как и любой язык программирования высокого уровня, поддерживает использование в основной программе подпрограмм (поименованных или иным образом идентифицированных частей программы). Подобные подпрограммы вызываются в соответствии со стандартом ANSI-C. Существует два основных вида подпрограмм:

1) Каждая подпрограмма, возвращающая только одно (собственное) значение, является функцией.

2) В том случае, если подпрограмма возвращает несколько значений, то согласно стандартам разработки программного обеспечения она является процедурой. Далее, в целях упрощения терминологии, и процедуры, и функции будут именоваться функциями.

Основная отличительная особенность функций заключается в том, что они создаются только один раз, после чего обращение к ним возможно из различных частей различных программ и/или функций (без необходимости повторного набора интересующих блоков кода). При вызове функции ей передаются значения переменных или констант, определенные в вызывающей программе или в вызывающей функции. Значения, передаваемые функции при ее вызове, именуется значениями «аргументов» (или значениями «фактических параметров»). То есть «аргументами» называются фактические переменные или выражения, значения которых используются как входные значения функции при ее вызове. Параметрами функции (формальными параметрами) именуется переменные в функции, которые будут содержать передаваемые в функцию входные значения.

Количество и последовательность фактических параметров (аргументов), указываемых при вызове функции, должны соответствовать количеству и последовательности формальных параметров, указанных в определении функции.

Более подробная информация о представленных выше взаимосвязях приведена в описании языка C. Пример ниже призван продемонстрировать только некоторую основную функциональность.

```
main() // Начало основной программы
{
float a, b, c; // Объявление переменных
```



```
a = 10; // Присвоение значений
```

```
b = 14;
```

```
c = sumFunction(a, b); /* Вызов функции с возвращаемым значением; передача значений двух аргументов */
```

```
scaleValues(a, b, c, 1.5); // Вызов функции
```

```
// с несколькими возвращаемыми значениями.
```

```
// 'a', 'b' и 'c' изменяются во время выполнения данной функции
```

```
DebugN(a, b, c); // Вывод результатов в средстве просмотра журналов
```

```
// a = 15, b = 21, c = 36
```

```
} // Окончание основной программы
```

```
/* Объявление функции с указанием типа данных возвращаемого значения и */
```

```
float sumFunction(float summand1, float summand2) // формальных параметров
```

```
{ // с указанием их типов
```

```
float sum;
```

```
sum = summand1 + summand2; // Применение формальных параметров
```

```
// в качестве переменных
```

```
return sum; // Возвращение значения (значения
```

```
// переменной 'sum') в вызывающую программу
```

```
}
```

```
// Объявление функции с несколькими возвращаемыми значениями
```

```
int scaleValues(float &a, float &b, float &c, float factor)
```

```
{ // '&' обозначает операцию получения адреса переменной: значения будут
```

```
a = a * factor; // возвращены
```

```
b = b * factor;
```

```
c = c * factor;
```

```
if ((a == 0) || (b == 0) || (c == 0)) // Опционально функция
```

```
return 0; // может возвращать некоторое значение,
```

```
else // например, код
```

```
return 1; // ошибки
```

```
}
```

Последовательность аргументов (фактических параметров) при вызове функции в точности соответствует последовательности формальных параметров в определении функции. Первая функция может быть представлена и в более компактной форме за счет использования вложенности выражений.

```
float sumFunction(float summand1, float summand2)
{
    return summand1 + summand2; // Вычисление и возврат возвращаемого значения
}
```

В примере выше возвращаемое значение функции "scaleValues()" никак не обрабатывалось в вызывающей программе. Обработка возвращаемого значения может быть реализована, например, следующим образом:

```
if (scaleValues(a, b, c, 1.5) > 0)
    DebugN(a,b,c);
else
    DebugN("Ошибка");
```

В данном примере возвращаемое значение сравнивается со значением проверочного выражения в рамках управляющей структуры "if".

В языке Control, наряду с представленными выше фактическими параметрами и формальными параметрами, существуют также и так называемые «параметры по умолчанию». Значения подобным параметрам присваиваются при их объявлении. Поддерживается также создание функций с произвольным количеством параметров.

Область видимости функции зависит от места ее определения:

- Область видимости в пределах сценария: при размещении определения функции ниже программы "main()" данная функция будет доступна только из программы "main()" и из всех других функций данного сценария.

- Область видимости в пределах панели: при размещении определения функции в тексте сценария события "ScopeLib" панели данная функция будет доступна для всех сценариев всех графических объектов панели.

– Область видимости в пределах менеджера: при размещении определения функции в отдельной библиотеке, физически расположенной в поддиректории `.../scripts/libs` директории проекта и загружаемой при запуске менеджера, данная функция будет доступна в пределах менеджера. Указание менеджеру на необходимость загрузки библиотеки осуществляется при помощи файла `"config.level."`.

## 7.7 Адресация элементов точек данных, конфигурационных элементов, атрибутов

При обращении из сценария к элементу точки данных, в сценарии необходимо указать полный идентификатор элемента точки данных. Идентификатор может состоять из нескольких уровней, количество которых определяется структурой точки данных.

Наряду с обращением к значению элемента точки данных, возможно обращение к ряду конфигурационных элементов их атрибутов.

`System1: Datapoint.element_1...element_n:_config.detail._attribute`

где

`System1` – имя системы SCAD CC (указание системы важно в случае адресации объекта в другой системе, входящей в состав распределенной системы SCAD CC, состоящей из нескольких взаимодействующих друг с другом систем SCAD CC);

`Datapoint` – имя точки данных (= имя устройства);

`element_1` – элемент(ы) структуры;

`element_n` – конечный элемент структуры — элемент, хранящий значение;

`config` – конфигурационный элемент (настройки функциональности элемента точки данных);

`detail` – уровень детализации (используется, например, при идентификации аларм-диапазонов);

`attribute` – атрибут конфигурационного элемента.

При обращении к значению элемента точки данных, в сценарии на языке Control достаточно указать идентификатор интересующего конечного элемента (элемента точки данных). При этом необходимость указания конфигурационного элемента и атрибута отсутствует. При наличии одной системы ее имя, указываемое в начале идентификатора, также можно опустить.

Идентификатор элемента точки данных обрабатывается в сценарии Control как последовательность символов (string). Для передачи строкового значения используются кавычки или переменная типа `"string"`.

При записи значения точки данных всегда необходимо обращаться к конфигурационному элементу `"_original"` (исходное значение). При чтении значения точки данных необходимо

обращаться к конфигурационному элементу "\_online" (текущее значение). При использовании сокращенной формы адресации данные особенности могут не приниматься во внимание, необходимые части идентификаторов (конфигурационные элементы и атрибуты) добавляются системой автоматически.

Запись или чтение значения при использовании сокращенной формы адресации

System1:Drive04.state.speed

Чтение значения

System1:Drive04.state.speed:\_online..\_value [anytype]

Запись значения

System1:Drive04.state.speed:\_original..\_value [anytype]

Время источника/ метка времени

System1:Drive04.state.speed:\_online..\_stime [time]

Состояние/ Качество сигнала

System1:Drive04.state.speed:\_online..\_status [bit32]

Инициатор (пользователь)

System1:Drive04.state.speed:\_online..\_user [uint]

Менеджер

System1:Drive04.state.speed:\_online..\_manager [uint]

Доступ к атрибутам других конфигурационных элементов осуществляется аналогичным образом. Ниже приведены несколько примеров. Имя системы опущено по причине адресации атрибутов в собственной системе. Типы данных заключены в кавычки.

Активация элемента точки данных

Drive04.state.speed:\_original..\_active [bool]

Достоверность значения

Drive04.state.speed:\_online..\_bad [bool]

Корректирующее значение

Drive04.state.speed:\_offline..\_corr [bool]

Нижняя граница диапазона [допустимых] значений

Drive04.state.speed:\_pv\_range..\_min [anytype]

Верхняя граница диапазона [допустимых] значений

Drive04.state.speed:\_pv\_range..\_max [anytype]

Цвет текущего аларма

Drive04.state.speed:\_alert\_hdl..\_act\_state\_color [string]

Тип данных "anytype" (выше) означает, что соответствующие атрибуты, в зависимости от различных факторов, поддерживают несколько типов данных (но не все, которые в принципе поддерживаются в системе).

## 7.8 Операции со значениями элементов точек данных

При обработке значений элементов точек данных в сценарии, значения элементов сначала должны быть считаны и записаны в локальные переменные. Далее в рамках сценария осуществляется необходимый расчет и преобразование, при этом для хранения промежуточных результатов обычно также используются переменные сценария. При необходимости результаты записываются в элементы точек данных. Наиболее важные функции:

```
dpGet(string dpe, anytype val, ...) // Считывание значений элементов точек данных
```

```
dpSet(string dpe, anytype val, ...) // Запись значений в элементы точек данных
```

Каждая из представленных выше команд может одновременно применяться к нескольким элементам точек данных. При выполнении каждой из команд доступ осуществляется однократно с возвратом соответствующего ответа непосредственно после вызова команды.

```
main()
```

```
{
```

```
float val1, val2;
```

```
bool val3;
```

```
string dpeXY;
```

```
dpGet("Drive04.state.speed", val1); // Чтение значения элемента точки данных
```

```
    // и запись считанного значения в переменную 'val1'
```

```
val1 = val2 + 20;
```

```
dpeXY = "Drive04.cmd.speed"; // Присвоение строковой переменной значения,
```

```
// соответствующего идентификатору элемента точки данных
```

```
dpSet(dpeXY, val2); // Запись в элемент точки данных
```

```
// 'Drive04.cmd.speed' значения
```

```
// переменной 'val2'
```

```
dpGet("Drive04.state.speed", val1, // Одновременное чтение значений нескольких элементов  
различных точек данных и запись считанных значений в переменные
```

```
    "Drive05.cmd.setpoint", val2,
```

```
    "Drive06.state.on", val3);
```

```
dpSet("Drive04.cmd.speed", val1, // Одновременная запись значений в несколько различных  
элементов различных точек данных
```

```
    "Drive05.cmd.setpoint", val2,
```

```
    "Drive06.cmd.on", !val3); // ! Инверсия значения
```

```
}
```

Элемент точки данных, к которому осуществляется доступ из сценария, должен существовать. Данное правило применимо в том числе и к конфигурационным элементам и атрибутам, поскольку данный конкретный элемент точки данных может содержать только ограниченный набор конфигурационных элементов из полного набора конфигурационных элементов, доступных в системе. Возникающие ошибки выводятся во время выполнения

программы, сохраняются в журнале и могут быть просмотрены при помощи средства просмотра журналов.

С целью повышения производительности рекомендуется объединение нескольких операций записи или чтения в рамках одного запроса.

В приведенном ниже сценарии, вызываемом при наступлении события "Clicked" кнопки, инвертируется режим (ручной <-> автоматический) клапана "V03". Значение элемента точки данных, соответствующего подаваемой команде, инвертируется при каждом нажатии кнопки левой кнопкой мыши. Соответствующее изменение графических свойств объектов на панели (экранной форме) выполняется симулятором.

```
main()

{

bool manualMode;

dpGet("V01.state.manual",manualMode);

dpSet("V01.cmd.manual",!manualMode); // '!' используется для инверсии значения

}
```

В примерах выше были рассмотрены функции однократного доступа. Однако во многих случаях требуется непрерывная актуализация информации, например, в экранных формах. Поскольку система SCAD CC является событийно-ориентированной, в системе доступны соответствующие функции.

```
main()

{

dpConnect(string callBackFunction,[bool noInit], string dpe [...]);

}
```

```
callBackFunction(string dpe, anytype value,...)

{

    // ... выполнение при изменении значения 'dpe'

}
```

Представленная выше функция обеспечивает подписку на изменения [атрибута] элемента точки данных 'dpe' в менеджере событий. При изменении значения оно незамедлительно возвращается в сценарий, при этом вызывается указанная функция обратного вызова. В приведенном ниже сценарии, вызываемом при наступлении события "Initialize" прямоугольника, в зависимости от состояния двоичного элемента точки данных "V03.state.manual" изменяется цвет фона прямоугольника.

```
main()

{

    dpConnect("showStateCB", "V03.state.manual");

}
```

```
showStateCB(string dpe, bool man)
```

```
// для каждого элемента точки данных, на изменение [атрибута] которого организуется подписка,
// должны быть заданы переменная типа 'string' для хранения идентификатора и
// переменная подходящего типа для хранения значения [атрибута] элемента
```

```
{
    if (man)
        this.backCol = "yellow"; // при 'man' == TRUE цвет фона изменяется на
        // желтый,
    else
        this.backCol = "grey"; // в противном случае цвет фона изменяется на серый
```



```
}
```

При помощи функции "dpConnect()" возможна одновременная подписка на изменения нескольких [атрибутов] элементов точек данных. При этом функция обратного вызова выполняется при изменении любого из указанных [атрибутов] элементов точек данных. При объявлении функции обратного вызова параметры необходимо указывать парами. Каждая пара параметров должна включать в себя переменную, соответствующую элементу точки данных, и переменную, соответствующую значению элемента точки данных.

## 7.9 Операции со значениями свойств графических объектов

Как правило, графические элементы поддерживают три интерфейса взаимодействия, позволяющих управлять внешним видом и функциональностью объектов, задавать действия графических объектов, а также получать информацию о внешних событиях в отношении объектов (требуемая реакция на данные события может быть определена в сценариях):

- Свойства: определяют внешний вид и функциональность, например, цвет, видимость и т.п.
- Методы: определяют действия графических объектов, например, сортировка таблицы.
- События: представляют собой флаги, информирующие о внешних событиях, например, о щелчке мышью, о получении фокуса и т.п. Требуемая реакция на данные события может быть определена в сценариях, запускаемых при наступлении событий.

Большинство свойств может быть прочитано или записано при помощи сценариев на языке Control. Принцип при выполнении операций со свойствами графических объектов схож с принципом выполнения операций с элементами точек данных: значения свойств считываются и присваиваются переменным, далее осуществляются необходимые преобразования, после чего актуализированные значения могут быть присвоены свойствам графических объектов.

Функция для чтения свойств:

```
getValue(string shape, string prop, anytype value [,anytype val2..]);
```

Подобным образом считывается и записывается в переменную "value" значение свойства 'prop' графического объекта "shape". В том случае, если свойство содержит два значения, необходимо добавить еще один параметр, например "scale" (масштаб) или "size" (размер).

```
main()  
{
```

```
string col;

getValue("Border1","backCol", col); // Прямоугольник с именем
// 'Border1': считывание
// свойства "Цвет фона" ('background color')
DebugN(col); // Вывод в средство просмотра журналов
}
```

В качестве альтернативы может использоваться сокращенная форма, принятая в C++. Например, утверждение (statement) для считывания значения цвета фона будет выглядеть при использовании сокращенной формы следующим образом:

```
col = Border1.backCol;
```

Для присвоения значения свойству при помощи сценария Control используется функция "setValue()":

```
setValue(string shape, string prop, anytype value [,anytype val2..]);
```

или

```
shape.prop = value;
```

При этом значение переменной "value" присваивается свойству "prop" графического объекта "shape". Сокращенная форма может также применяться в сценариях, выполняемых при наступлении событий данного графического объекта:

```
setValue("", "prop", val);
```

```
getValue("", "prop", val);
```

```
this.prop = val;
```

```
val = this.prop;
```

Оба варианта обеспечивают один и тот же результат. Аналогично элементам точек данных, в рамках одной команды поддерживается одновременное чтение или одновременная запись нескольких свойств одного графического объекта. В данном случае необходимо использовать полную форму с использованием "getValue()" или "setValue()".

```
setValue("", "backCol", stringVar1, "text", textVar1);
```

Одновременное чтение или одновременная запись нескольких свойств нескольких графических объектов возможны при использовании функций "getMultiValue()" и "setMultiValue()".

В том случае, если символ устройства состоит из нескольких графических объектов, для управления графическими свойствами рекомендуется использовать отдельный сценарий. В том случае, если сценарий должен запускаться при щелчке ЛКМ по любой точке в пределах интересующей области, то сценарий может быть привязан к событию "Clicked" нескольких прозрачных объектов, например, прямоугольников, нарисованных поверх интересующей области. При этом прямоугольники должны иметь прозрачные ("\_Transparent") цвета фона и рисунка. После добавления следующей строки в сценарии для события "Initialize" некоторого объекта заливка данного объекта изменяется на сплошную, и событие "Clicked" будет наступать при щелчке левой кнопкой мыши и в пределах данного объекта:

```
this.fill = "[solid]";
```

## 7.10 Менеджеры сценариев

Сценарии, созданные на языке Control, могут выполняться не только менеджерами интерфейса пользователя, но также и менеджерами сценариев. При этом обеспечиваются практически те же возможности, что и при выполнении сценариев менеджерами интерфейса пользователя. Основное отличие заключается в том, что в сценариях, выполняемых менеджерами сценариев, недоступны функции для работы со свойствами графических объектов и функции для работы с панелями. Менеджер сценариев взаимодействует с другими менеджерами SCAD CC, используя элементы точек данных (т.е. через образ процесса).

Сценарии хранятся в директории "<путь\_проекта>.scripts/" в виде ASCII-файлов. В проекте "GettingStarted" сценарии, выполняемые менеджерами сценариев, используются для симуляции реакций устройств и для симуляции процесса.

Любой файл сценария (\*.ctl), предназначенный для выполнения менеджером сценариев, содержит как минимум функцию "main()", включающую в себя ряд событийно-ориентированных команд ("dpConnect()" с соответствующими функциями обратного вызова). Также поддерживается возможность использования периодических процессов (конструкций с операторами "while", "do" и с задержкой цикла).

В том случае, если сценарий состоит только из набора присвоений значений и не содержит подписок на изменения значений ("dpConnect()", "dpQueryConnect()", ...) или бесконечных циклов, обработка сценария будет завершена после выполнения последнего присвоения. В том случае, если данный сценарий является единственным сценарием, выполняемым данным менеджером

сценариев, то по завершении сценария работа менеджера также будет прекращена. Непрерывное выполнение сценария должно быть обеспечено разработчиком сценария.

Функции, вызываемые из функции "main()", могут быть объявлены непосредственно после текста функции "main()" (так же как и в сценариях, связанных с событиями графических объектов). В результате они будут доступны только из функции "main()", а также из всех других локально объявленных функций.

Функции также могут быть объявлены глобальными. Для этого функции должны быть объявлены в отдельных библиотеках, представляющих собой ASCII-файлы. Данные библиотеки должны располагаться в директории "<путь\_проекта>/scripts/libs/". Объявление библиотек обеспечивается путем использования соответствующей записи в конфигурационном файле "config.level" или путем использования ключевого слова "#uses" в текстах сценариев.

Переменные с областью видимости в пределах сценария объявляются в сценарии перед функцией "main()". В том случае, если переменные должны быть доступны из всех сценариев, выполняемых данным менеджером, они должны быть объявлены в библиотеке до объявления функций. Глобальные переменные также могут быть объявлены в режиме выполнения программы при помощи функции "addGlobal()" языка Control.

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**

<b>Сокращение</b>	<b>Расшифровка</b>
CSV	Comma-Separated Values
DB	Database
DDE	Dynamic Data Exchange
DIST	Distribution Manager – Менеджер распределенной архитектуры
DM	Data Manager – Менеджер базы данных, Менеджер БД
DP	Data point – Точка данных
DPA	Data point attribute – Атрибут точки данных
DPE	Data point element – Элемент точки данных
DPT	Data point type – Тип точки данных
EV	Event Manager – Менеджер событий («диспетчер данных»)
GEDI	Graphic Editor GEDI – Графический редактор
HMI	Human Machine Interface – Человеко-машинный интерфейс, ЧМИ
HTTP	HyperText Transfer Protocol
IEC	International Electrotechnical Commission
IP	Internet Protocol
LV	Log Viewer – Средство просмотра журналов (системный журнал)
MP	Master data point
OLE-DB	Object Linking and Embedding-Database
OPC	Open Platform Communications
PARA	Инструмент для конфигурирования
PLC	Programmable logic controller – Программируемый логический контроллер, ПЛК
PMON	Process Monitor – Монитор процессов (мониторинг системы)
REDU	Redundancy Manager – Менеджер резервирования (подключение резервирующей системы)
RTU	Remote Terminal Unit, Remote Telemetry Unit – Устройство связи с объектом, УСО
SCADA	Supervisory Control And Data Acquisition

Сокращение	Расшифровка
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSI	Server Side Includes – Включения на стороне сервера
TCP	Transmission Control Protocol
UI	User Interface – Интерфейс пользователя (визуализация)
VA	Value Archive – Архив значений (БД значений)
VISION	Модуль визуализации
АРМ	Автоматизированное рабочее место
БД	База данных
ИБП	Источник бесперебойного питания
ЛКМ	Левая кнопка мыши
МЭК	Международная электротехническая комиссия
ОЗУ	Оперативная память
ПКМ	Правая кнопка мыши
ПЛК	Программируемый логический контроллер
ПО	Программное обеспечение
РБД	Реляционная база данных
УСО	Устройство связи с объектом