

ООО «КАСКАД-АСУ»

SCADA-система «КАСКАД».

Курс лекций
для студентов высших учебных заведений

Чебоксары

2019

Оглавление

Лекция 1. SCADA-системы. КАСКАД. Общие сведения. Конфигуратор.....	3
1. Введение	3
2. Конфигуратор.....	6
Лекция 2. Пользователи и пароли. События. СДД. Паспорта.	8
1. Пользователи и пароли	8
2. События	11
2.1. Модуль настройки баз данных событий	11
2.2. Модуль регистрации событий	13
2.3. Модуль просмотра событий	13
3. Сервер Доступа к Данным	14
4. Паспорта.....	16
Лекция 3. Система программирования микропроцессорных контроллеров с открытой архитектурой KLogic.....	19
1. Понятие технологической программы.....	19
2. Структура KLogic.....	19
3. Архитектура исполнительной системы KLogic.....	22
Лекция 4. Клиентские модули SCADA-системы «КАСКАД»	24
1. Модуль Визуализации	24
2. Модуль регистрации технологических параметров.....	29
2.1. Модуль настройки БД ТП.....	26
2.2. Модуль регистрации значений ТП в БД.....	28
2.3. Модуль извлечения данных из БД ТП.....	28
3. Модуль формирования рапортов.	31
4. Модуль просмотра истории	33
Лекция 5. Модуль аварийной и предупредительной сигнализации.....	37
1. Общие сведения	37
2. Настройка модуля АПС.	38
3. Распределенные системы сигнализации.....	41

Лекция 1. SCADA-системы. КАСКАД. Общие сведения. Конфигуратор

1. Введение.

Современная промышленность немыслима без систем автоматизации. Автоматика применяется повсеместно: в управлении и контроле технологическим процессом, в контроле и дозировании ресурсов, в учете потребляемой и вырабатываемой энергии.

В системах промышленной автоматизации широко используются электронно-управляемые и интеллектуальные устройства: задвижки, клапана, датчики, счетчики, расходомеры, термометры и т.д.

Информация с этих устройств собирается и обрабатывается локальными интеллектуальными устройствами – контроллерами, вычислителями, устройствами сбора и передачи данных (УСПД). На этом уровне производится первичная обработка поступивших данных, производится локальное управление: регулирование, контроль защит, выполнение алгоритмов.

Информация с контроллеров передается на сервера, обрабатывается, сводится, архивируется, анализируется и передается на автоматизированные рабочие места — АРМы, где и выдается оператору. Оператор, таким образом, получает информацию обо всем технологическом процессе, насколько бы распределенным он ни был.

Программный пакет, позволяющий осуществить такую обработку, и называется *SCADA-системой*.

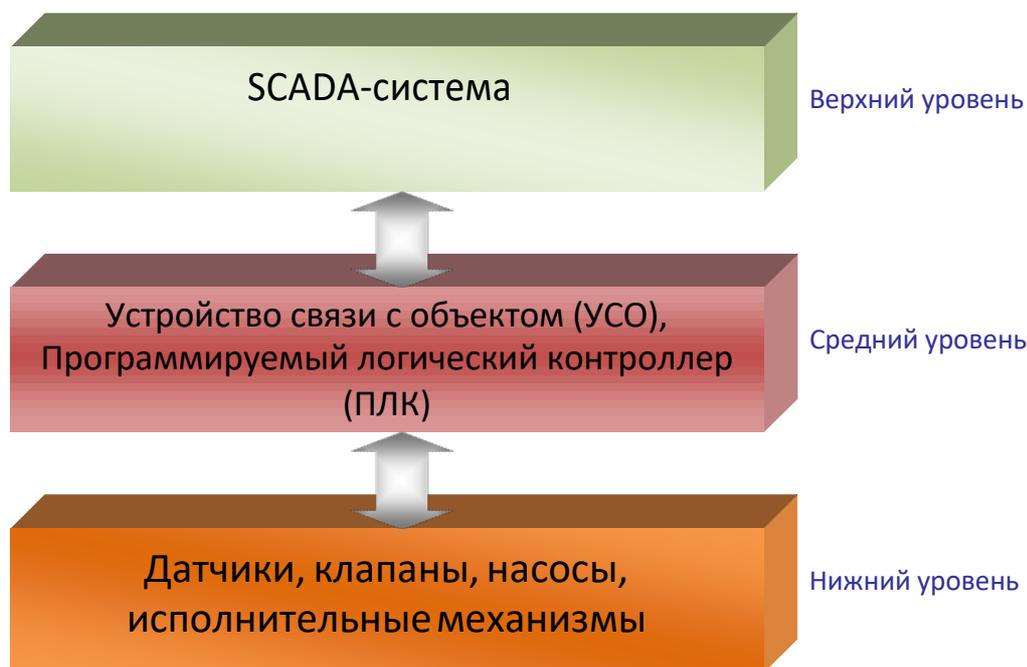


Рис. 1.1. Уровни систем промышленной автоматизации.

SCADA (аббр. от англ. *Supervisory Control And Data Acquisition*, Диспетчерское управление и сбор данных) — программный пакет для сбора, обработки, отображения и архивирования информации об объекте управления.

Одним из представителей программных продуктов этого класса является SCADA-система «КАСКАД», которая предназначена для построения различных систем промышленной автоматизации, например:

- **АСУТП** – автоматизированная система управления технологическим процессом
- **АСКУЭ** – автоматизированная система контроля и учета электрической энергии

- **АСКУЭР** – автоматизированная система контроля и учета энергоресурсов
- **АСДКУ** – автоматизированная система диспетчерского контроля управления и др.

SCADA-система «КАСКАД» выполняет следующие основные функции:

- Сбор оперативной информации о ходе технологического процесса
- Обработка полученной информации по алгоритмам пользователя
- Предоставление информации в виде мнемосхем технологического процесса
- Оперативное и диспетчерское управление технологическим процессом
- Ведение истории технологического процесса
- Вычитка накопленных архивов устройств (контроллеров, счетчиков)
- Просмотр и анализ хода технологического процесса
- Формирование отчетной документации
- Экспорт оперативной и исторической информации в WEB
- Регистрация всех действий операторов, а также контроль и регистрация событий и нарушений в ходе технологического процесса
- Информирование персонала о нарушениях при помощи звуковой, световой сигнализации, через короткие сообщения (SMS) и по электронной почте (E-mail)
- Разграничение прав пользователей и уровней доступа к системе

«КАСКАД» — модульная система, имеющая клиент-серверную архитектуру. Можно выделить две основные группы программных модулей SCADA-системы «КАСКАД» — серверные и клиентские. Серверная часть отвечает за сбор данных с различных устройств, их обработку, ведение архивов, отслеживание нарушений в технологическом процессе. К ней относятся следующие модули:

- *Сервер Доступа к Данным (СДД)*
- *Модули Доступа к Данным (МДД)* – драйвера, реализующие протоколы обмена с различными устройствами: как стандартные (Modbus, OPC), так и специфичные для конкретных устройств
- *Модули Обработки Данных (МОД)* – модуль обработки паспортов, модуль аварийно-предупредительной сигнализации, менеджер опроса контроллеров KLogic, модуль ведения баз данных (БД) технологических параметров (ТП)

Клиентская часть обеспечивает взаимодействие с пользователем: наблюдение за ходом технологического процесса, управление процессом, просмотр накопленных архивов, формирование отчетов. К ним относятся следующие модули:

- *Модуль Визуализации*
- *Модуль просмотра истории*
- *Модуль формирования рапортов*

Кроме того, в **«КАСКАД»** интегрированы две обязательные подсистемы, которые используются всеми приложениями — и серверными, и клиентскими — это подсистема аутентификации пользователей, и подсистема регистрации событий.

Подсистема *аутентификации пользователей* необходима для исключения несанкционированного доступа к системе. Кроме того, она позволяет дать пользователям разные права: одним пользователям разрешено только слежение за технологическим процессом, другим управление, третьим – изменение тонких параметров технологического процесса и уставок.

Задачей подсистемы *регистрации событий*, как следует из названия, является регистрация всех событий, происходящих в системе: действия пользователей, выход за допустимые пределы параметров технологического процесса, сработка защит, а также иные события, генерируемые приложениями.

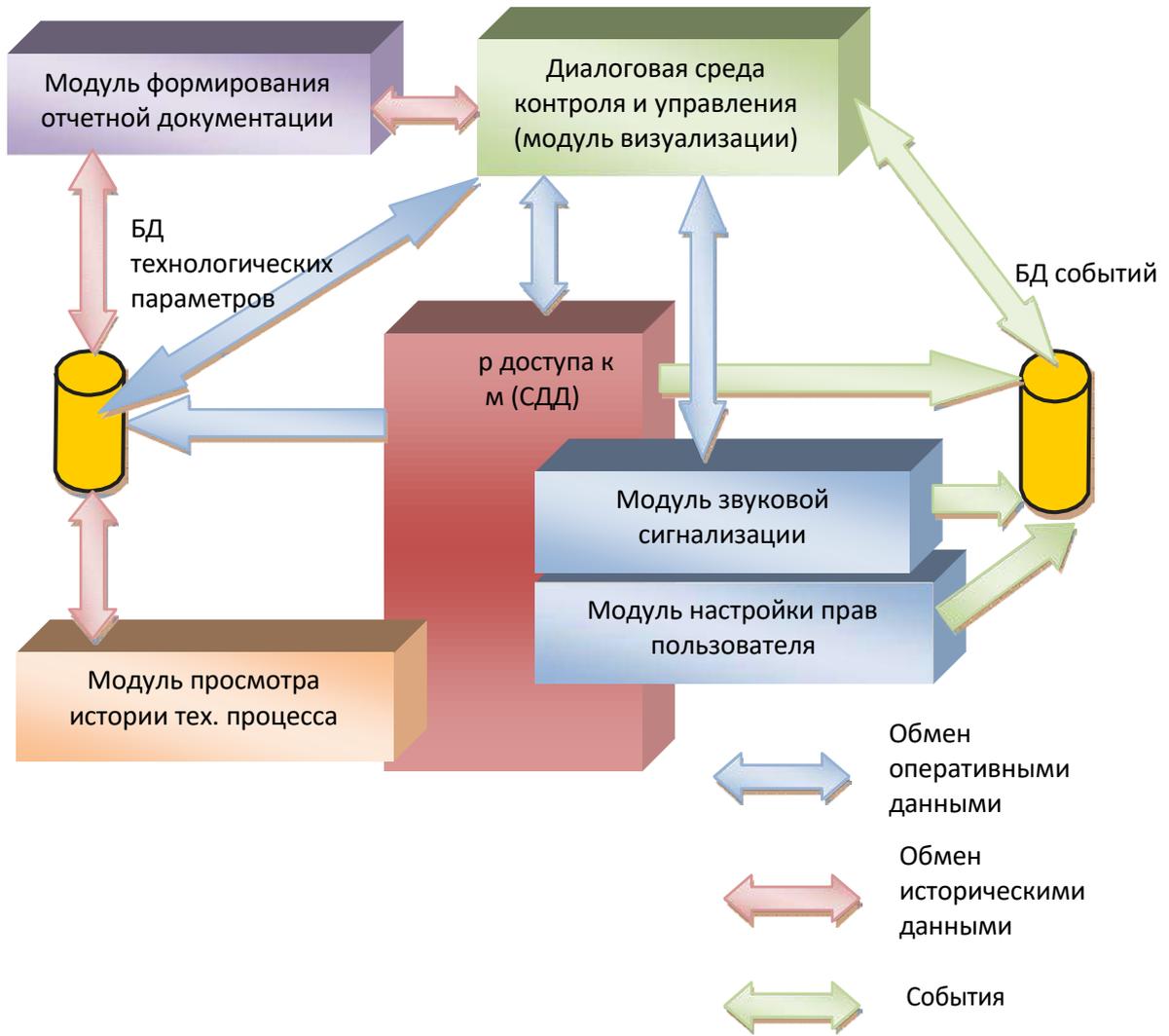


Рис. 1.2. Модули SCADA-системы «КАСКАД».

2. Конфигуратор.

Назначение Конфигуратора – свести все модули SCADA-системы «КАСКАД» в единую оболочку, позволяющую удобно создавать и настраивать проекты.

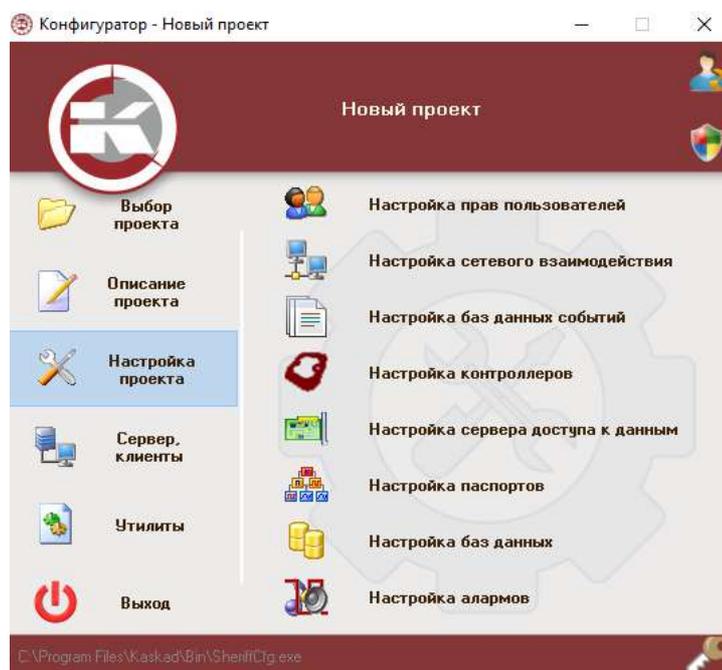


Рис. 2.1. Конфигуратор.

Конфигуратор имеет пять вкладок: выбор проекта, описание проекта, настройка проекта, сервер и клиенты, утилиты.

- Вкладка **Выбор проекта** позволяет открыть существующий проект, либо создать новый, запустив мастер создания проектов.
- Вкладка **Описание проекта** выводит информацию об открытом проекте
- На вкладке **Настройка проекта** расположены кнопки вызова настроечных модулей для всех подсистем SCADA-системы «КАСКАД»
- На вкладке **Сервер, клиенты** расположены кнопки вызова исполняемых модулей SCADA-системы «КАСКАД»
- С вкладки **Утилиты** запускаются вспомогательные утилиты SCADA-системы.

Все кнопки расположены в той последовательности, в которой должна логически происходить настройка проекта:

- 1) **Настройка прав пользователей.** Поскольку любые изменения в проект разрешено вносить только пользователям, прошедшим аутентификацию, в первую очередь создается база данных пользователей и сами пользователи, им назначаются соответствующие права (для начала достаточно одного пользователя с правами администратора).
- 2) **Настройка сетевого взаимодействия.** Следующий этап – создание сетевых станций. Каждая сетевая станция – это сервер, который будет собирать данные с устройств и передавать им управляющие воздействия. Для начала достаточно создать хотя бы одну сетевую станцию и назначить ей адрес. Пусть это будет 127.0.0.1.
- 3) **Настройка баз данных событий.** Все события, генерируемые приложениями, должны записываться в базу данных событий. Без этой базы ни одно из приложений SCADA-системы «КАСКАД» не запустится. Следовательно, создаем эту базу.
- 4) Настраиваем наши источники данных. Пункт **Настройка контроллеров** запустит инструментальную часть системы программирования KLogic, в которой можно создать

технологическую программу для контроллера или контроллеров – как реальных, так и виртуальных.

- 5) Источником данных может быть не обязательно контроллер под управлением KLogic. Альтернативный вариант получения данных, который может работать параллельно предыдущему – это опрос через модули доступа к данным. Конфигуратор этих модулей вызываем через пункт **Настройка сервера доступа к данным**. В нем выбираем нужные драйвера для опроса устройств и создаем теги для тех сигналов, которые нам будут нужны для работы.
- 6) Для того, чтобы теги устройств стали доступны остальным приложениям и подсистемам «КАСКАДа», а также для возможной дополнительной обработки сигналов необходимо создать **паспорта**. Произвести такую настройку нам поможет **Настройка паспортов**.
- 7) Настроенные паспорта и параметры KLogic – это те конечные данные, которые пользователь будет видеть на экране. Их достаточно для получения оперативной картины технологического процесса. Но для того, чтобы хранить историю технологического процесса, необходимо настроить базы данных. Конфигуратор баз данных ТП вызываем при помощи пункта **Настройка баз данных технологических параметров**.
- 8) И завершающий этап — **Настройка алармов**. Здесь мы конфигурируем модуль аварийно-предупредительной сигнализации: указываем, за какими параметрами необходимо следить, по каким критериям оценивать нарушение технологического процесса, каким способом производить уведомление оператора: при помощи цветовой, звуковой сигнализации, либо рассылкой смс или сообщений электронной почты.

После выполнения всех этих пунктов мы получаем готовый настроенный сервер, способный опрашивать устройства, вести базы данных, выдавать информацию клиентским приложениям и сигнализировать о нарушениях ТП.

Далее рассмотрим каждый из пунктов подробнее.

Лекция 2. Пользователи и пароли. События. СДД. Паспорта.

1. Пользователи и пароли.

SCADA-система собирает информацию о технологическом процессе, предоставляет ее пользователю в удобном для анализа виде, позволяет вносить изменения в ход технологического процесса и управлять им. Вполне очевидно, что не вся информация предназначена для посторонних глаз, не говоря уже о том, что управление технологическим процессом разрешено строго определенному кругу лиц. Следовательно, SCADA-системе необходима некая подсистема, позволяющая опознать пользователя и разрешить или запретить доступ к той или иной информации и/или управлению технологическим проектом.

В SCADA-системе «Каскад» эти функции выполняет **Подсистема аутентификации пользователей**. Конфигуратор этой подсистемы запускается из пункта **Настройка прав пользователей**.

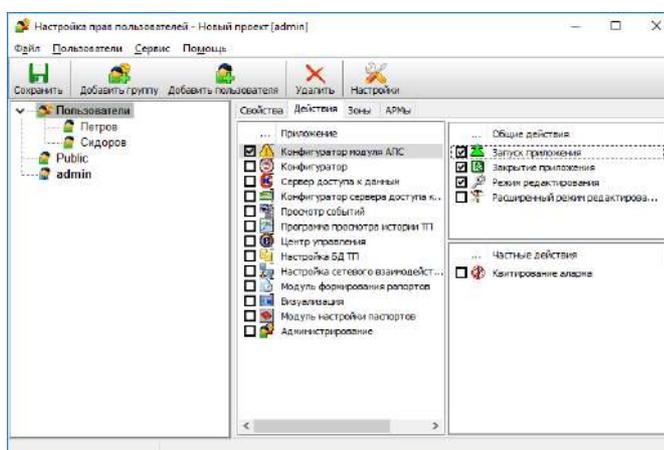


Рис. 1.1. Настройка прав пользователей.

Конфигуратор настройки прав пользователей позволяет добавлять в SCADA-систему отдельных пользователей и их группы, назначать права управления различными приложениями.

Если планируется создать несколько пользователей с одинаковыми правами, то удобнее всего создать группу, назначить ей права, а уже потом добавлять в нее пользователей. Каждый пользователь в группе в этом случае будет обладать теми же правами, что и группа. Плюс ко всему, отдельному пользователю можно дополнительно добавить или ограничить некоторые права.

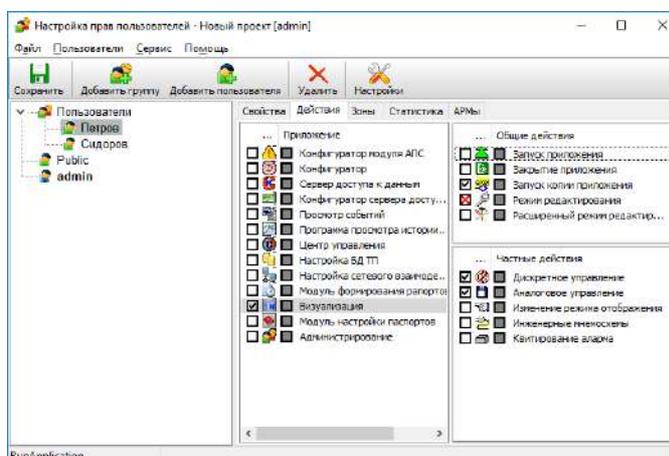


Рис. 1.2. Назначение дополнительных прав пользователю в группе.

На рис. 1.1 видно, что пользователям группы «Пользователи» разрешено запускать модуль визуализации, выгружать его, а также переводить его в режим редактирования. Однако пользователю «Иванов» разрешены помимо этого еще и аналоговое и дискретное управление, а переход в режим редактирования запрещен.

Все настройки прав пользователей хранятся в базе данных пользователей. Каждое из приложений «Каскада» должно иметь доступ к этой базе, иначе приложение просто не запустится.

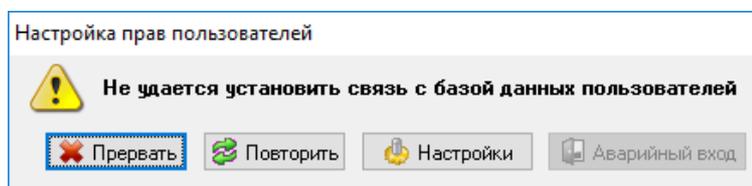


Рис. 1.3. При отсутствии доступа к БД пользователей приложение не запустится.

Настроить связь с базой данных можно через пункт **Настройки** (рис. 1.4.). Имя пользователя и пароль в данном случае – это имя пользователя и пароль для доступа к СУБД Firebird, в которой хранится информация о пользователях. По умолчанию это SYSDBA и masterkey. Разумеется, для более надежной защиты системы имя пользователя и пароль необходимо изменить.

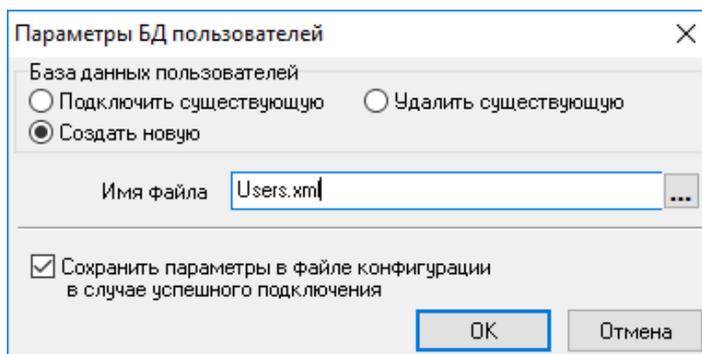


Рис.1.4. Настройка соединения с базой данных пользователей.

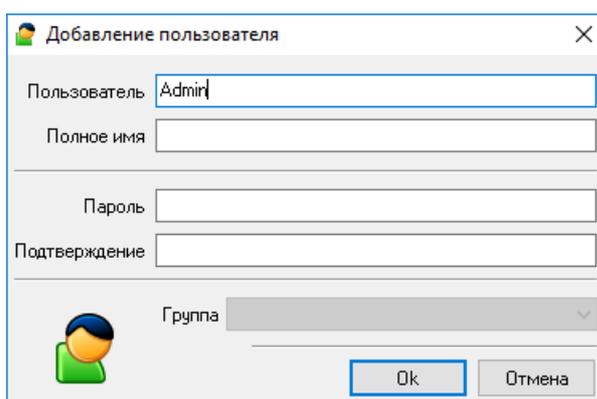


Рис.1.5. Добавление пользователей.

Любой модуль SCADA-системы «Каскад» при запуске обязательно должен подключиться к базе данных пользователей, идентифицировать пользователя, запустившего этот модуль, и проверить его права на выполнение тех или иных действий. Этими действиями, общими для всех модулей SCADA-системы «Каскад», являются:

- запуск приложения (разрешение на запуск приложения)
- закрытие приложения (разрешение на закрытие приложения)
- запуск копии приложения (разрешение на запуск копии приложения, если приложение позволяет запускать несколько копий)

- режим редактирования (разрешение на переход в режим редактирования без удаления или создания новых объектов)
- расширенный режим редактирования (разрешение на переход в расширенный режим редактирования, включая удаление и создание новых объектов).

Помимо этого, различные модули SCADA-системы «Каскад» могут иметь свои, частные действия, определяемые спецификой приложения. Например, для модуля визуализации это

- Дискретное управление – управление дискретными сигналами
- Аналоговое управление – управление аналоговыми сигналами
- Изменение режима отображения – изменение стиля отображения мнемосхем
- Квитирование аларма – разрешение на снятие звукового сигнала при возникновении сигнализации

Инженерные мнемосхемы – разрешение на доступ к специальным мнемосхемам, имеющим статус «инженерные».

Главный принцип подсистемы аутентификации прав пользователей: запрещено все, что не разрешено. То есть по умолчанию пользователю запрещены все действия, на которые ему не назначены разрешения. И, разумеется, если пользователю назначены права администратора, то ему разрешено все.

2. События.

Подсистема регистрации событий является обязательным компонентом SCADA-системы «КАСКАД». Её основная функция — это регистрация всех событий, происходящих в системе. Под событиями понимаются действия пользователей, изменение настроек, управление технологическим процессом, аварийная сигнализация и реакция оператора на неё, сработка защит, служебные логи серверных модулей системы и т.д.

Используемая СУБД для баз данных событий – Firebird.

Подсистема регистрации событий состоит из следующих частей:

- 1) модуль настройки баз данных событий;
- 2) модуль регистрации событий;
- 3) модуль просмотра событий.

2.1. Модуль настройки баз данных событий.

2.1.1. Базы данных событий.

Для любого проекта обязательно должна быть настроена хотя бы одна база данных событий, иначе система не будет работать. Создавать новые проекты следует из Конфигуратора, с помощью «Мастера создания проектов», («Настройка проекта» -> «Настройка баз данных событий») который позаботится о наличии в новом проекте как БД пользователей, так и БД событий. Созданная БД событий будет являться **БД по умолчанию** – базой данных, которая используется всеми станциями проекта (в том числе незарегистрированными) и которую нельзя удалить.

В простейшем случае для проекта достаточно настроить только одну базу данных событий. Если необходимо, существует возможность создавать и вести свои БД событий для каждой станции или для группы станций проекта. При этом каждая станция будет производить запись событий в соответствующую ей базу данных. Если станция не зарегистрирована в проекте (с помощью программы «Настройка сетевого взаимодействия»), то для неё будет использоваться **БД по умолчанию**.

Любая (но только одна) БД событий проекта может быть назначена в качестве **БД по умолчанию**. В списке БД событий **БД по умолчанию** выделена жирным шрифтом:

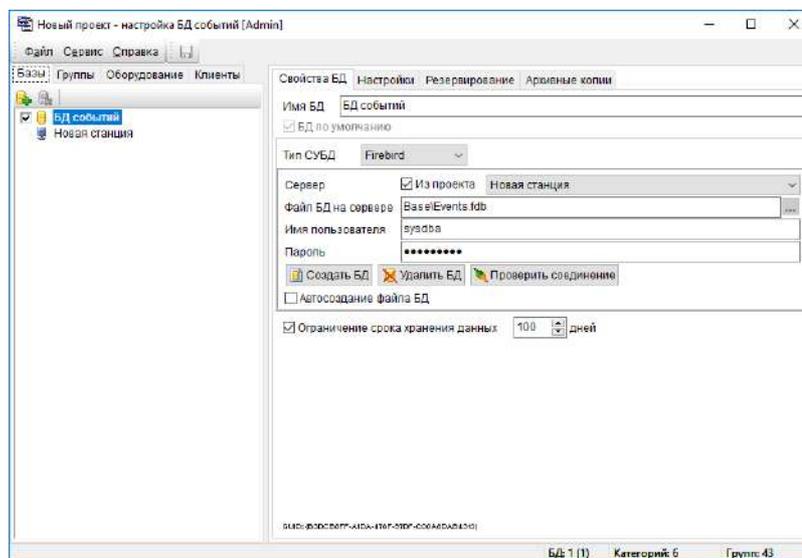


Рис. 2.1. Настройка БД событий

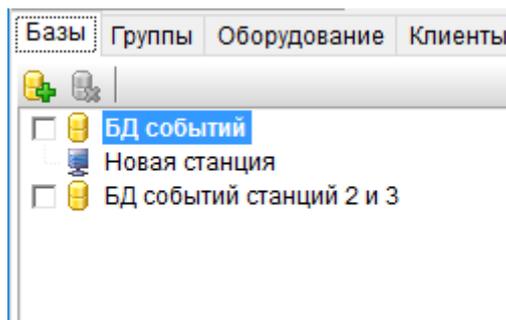


Рис. 2.2. Список БД событий

Для каждой БД событий обязательно необходимо настроить параметры подключения к файлу БД и создать собственно файл БД. Кроме этого, имеется возможность задать ограниченный срок хранения данных, настроить резервирование и создание архивных копий, посмотреть статистику по выбранной БД, создать backup-копию и т.д.

2.1.2. Группы событий.

Для того, чтобы визуально отличить события различного происхождения и важности друг от друга, каждое событие системы записывается в свою группу. Группы событий могут объединяться в категории. Количество групп и категорий в проекте не ограничено.

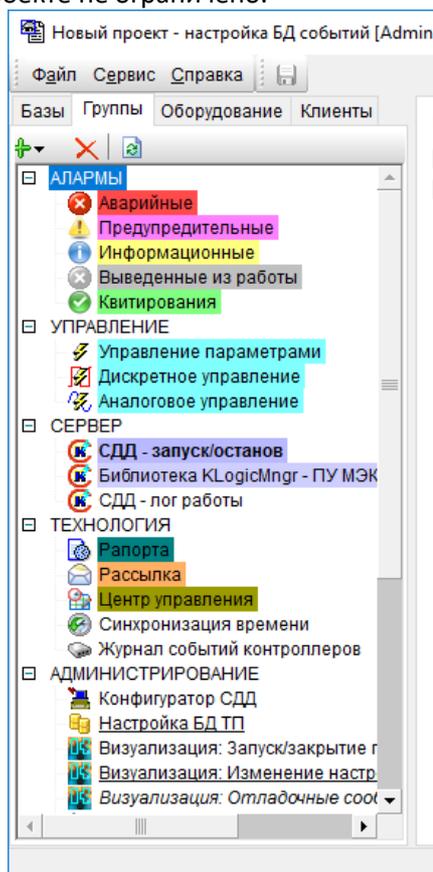


Рис. 2.2. Категории и группы событий

На рис. 2.2 корневыми элементами списка являются категории, дочерними – группы.

«Мастер создания проектов» автоматически добавляет в проект список групп событий по умолчанию. При необходимости этот список можно изменить, добавив туда новые категории и группы.

Кроме того, новые группы событий могут автоматически создаваться модулями SCADA-системы в процессе её работы. Такие группы по умолчанию будут относиться к категории, помеченной «для новых групп». При необходимости их можно переместить в любую другую категорию.

Для визуального выделения событий, относящихся к той или иной группе, в свойствах группы можно настроить цвет шрифта и фона, а также выбрать подходящую пиктограмму. Таким образом события «особо важных групп» можно выделить в общем списке ярким цветом или фоном.

Как правило, группы событий невозможно удалить из проекта, т.к. большинство групп создаётся модулями SCADA-системы автоматически. Из проекта возможно удалить только группы, добавленные пользователем. Тем не менее, любую группу или даже целую категорию событий при необходимости можно отметить как «незаписываемую». События, относящиеся к подобным группам, будут игнорироваться модулем регистрации событий и, соответственно, не будут записываться в БД.

2.2. Модуль регистрации событий

Как следует из названия, данный модуль производит регистрацию всех событий системы в базы данных событий – это его основная функция. Кроме этого, он осуществляет очистку БД от устаревших данных и резервирование, если это настроено.

2.3. Модуль просмотра событий

Этот модуль служит для просмотра событий SCADA-системы «КАСКАД» в виде списка.

Клиент	Т события	БД	Группа	Событие
192.168.1.233	10:26:34	БД событий	Библиотека KLogicMgr	МЭК-соединение с контроллером "Vir1_contr"
192.168.1.233	10:26:34	БД событий	Библиотека KLogicMgr	МЭК-соединение с контроллером "Контролл"
192.168.1.233	10:21:53	БД событий	СДД - лог работы	==== Завершен переход в режим работы. ==
192.168.1.233	10:20:41	БД событий	СДД - лог работы	СДД: Конфигурация модуля не найдена (Cf
192.168.1.233	10:20:37	БД событий	Библиотека KLogicMgr	МЭК-соединение с контроллером "Vir1_contr"
192.168.1.233	10:20:36	БД событий	СДД - лог работы	----- Перевод сервера в режим работы.
192.168.1.233	10:20:36	БД событий	СДД - лог работы	Продвижение работы СДД была завершена н
192.168.1.233	10:20:36	БД событий	Библиотека KLogicMgr	МЭК-соединение с контроллером "Контролл
192.168.1.233	10:20:35	БД событий	СДД - лог работы	Инициализация - ОШИБКА
192.168.1.233	10:20:35	БД событий	СДД - лог работы	Ошибка инициализации: Для сервера не най
192.168.1.233	10:20:35	БД событий	СДД - лог работы	Загрузка - УСПЕШНО
192.168.1.233	10:20:34	БД событий	СДД - лог работы	Серверный модуль доступа к вторичным БД
192.168.1.233	10:20:34	БД событий	СДД - лог работы	Инициализация - ОШИБКА
192.168.1.233	10:20:34	БД событий	СДД - лог работы	Ошибка инициализации: В проекте нет ни о
192.168.1.233	10:20:34	БД событий	СДД - лог работы	Загрузка - УСПЕШНО
192.168.1.233	10:20:33	БД событий	СДД - лог работы	Инициализация - УСПЕШНО
192.168.1.233	10:20:33	БД событий	СДД - лог работы	Модуль ведения вторичных БД TP D Maskad
192.168.1.233	10:20:32	БД событий	СДД - лог работы	Инициализация - УСПЕШНО
192.168.1.233	10:20:32	БД событий	СДД - лог работы	Модуль регистрации в БД ТП простого форм.
192.168.1.233	10:20:29	БД событий	СДД - лог работы	Загрузка - УСПЕШНО
192.168.1.233	10:20:29	БД событий	СДД - лог работы	Модуль аварийной и предупредительной си
192.168.1.233	10:20:29	БД событий	СДД - лог работы	Загрузка - УСПЕШНО

Рис. 2.3. Список событий

Модуль просмотра событий имеет следующие возможности:

- фильтрация событий по выбранным базам данных, группам событий, зонам производства, станциям, типам событий;
- выборка событий за конкретную дату или период;
- выделение событий по группе, зоне производства или даже по параметру;
- следящий режим;
- поиск и сортировка событий;
- предварительный просмотр и печать;
- экспорт событий в файл Microsoft Excel, в текстовый файл или в отдельный файл БД.

Дополнительно к модулю просмотра событий, в SCADA-системе «КАСКАД» существуют панель расширения и объект для модуля визуализации, предназначенные для вывода событий на мнемосхемы. Кроме этого, в модуле формирования рапортов имеется специальный алгоритм, позволяющий генерировать отчёты, содержащие данные из БД событий.

3. Сервер Доступа к Данным.

Сервер Доступа к Данным, или СДД,- это одно из основных приложений SCADA-системы «Каскад». Задачей его является сбор и обработка данных с устройств, архивирование данных в соответствии с заданными алгоритмами, контроль данных и генерация событий сигнализации (алармов) в случае выхода параметров за допустимые пределы.

За выполнение каждой из задач отвечает свой модуль в составе СДД. По сути, сам СДД – это некая оболочка, объединяющая различные модули и обеспечивающая их взаимодействие между собой в рамках заданной роли. Благодаря такой модульной организации можно легко дополнить функционал СДД без необходимости его перекомпиляции. Любой сторонний пользователь может написать свой драйвер или модуль обработки данных, который автоматически будет подключен к Серверу Доступа к Данным. Разумеется, при этом необходимо обеспечить наличие необходимых программных интерфейсов в модуле, но рассматривать их в рамках данной лекции мы не будем.

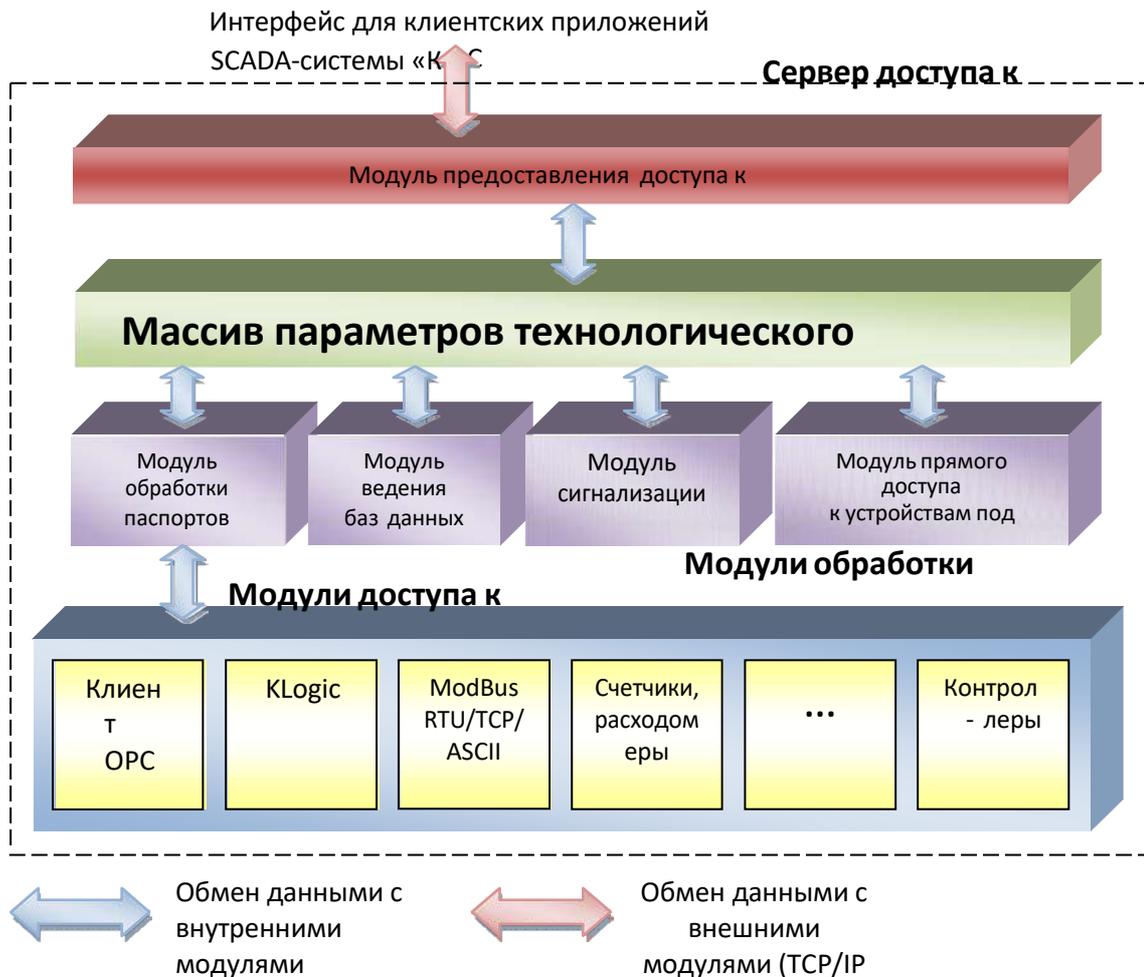


Рис. 3.1 Структура Сервера Доступа к Данным.

Настройка модулей доступа к данным (МДД) и некоторым модулям обработки данных (МОД) производится в **Конфигураторе Сервера Доступа к Данным**.

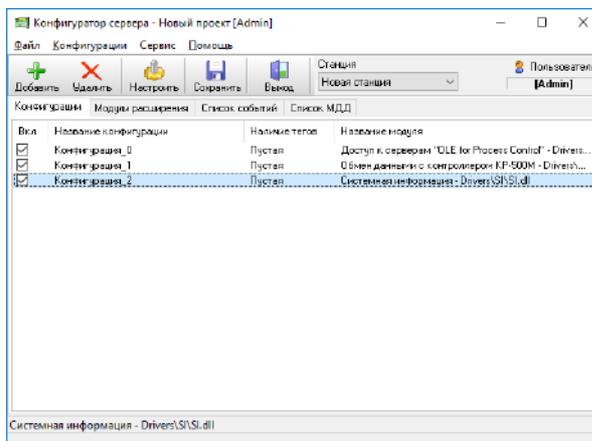


Рис.3.2. Конфигуратор СДД.

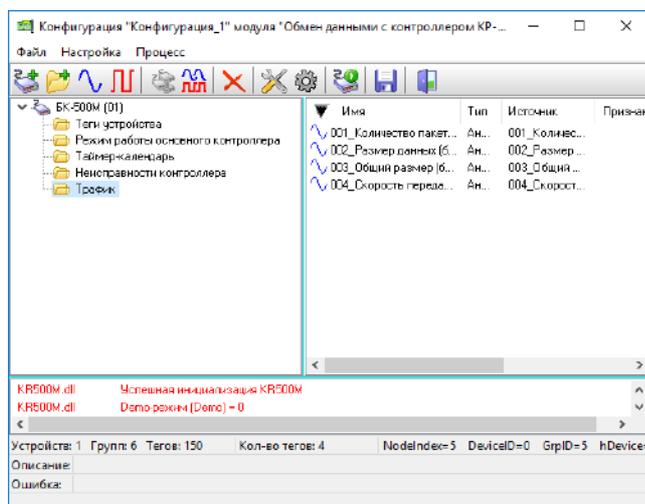
Для каждого задействованного в обработке данных МДД создается и настраивается своя конфигурация. Список настроенных конфигураций доступен на вкладке **Конфигурации**. Полный список доступных МДД отображается на вкладке **Список МДД**.

Наличие галочки около названия конфигурации говорит о том, что при загрузке СДД указанная конфигурация загрузится и начнет работать. Отсутствие галочки, соответственно, укажет на то, что данная конфигурация в работу СДД включена не будет.

При двойном щелчке на конфигурации или нажатии на кнопку **Настроить** откроется окно настройки конфигурации.

Список задействованных МОДов доступен на вкладке **Модули расширения**. Модули расширения могут быть обязательными и необязательными. Обязательные – это модуль обработки паспортов, модуль регистрации технологических параметров, модуль аварийно-предупредительной сигнализации. Настройка этих МОДов производится соответствующими внешними приложениями отключить их нельзя, поэтому в списке модулей расширения они отсутствуют. Остальные МОДы настраиваются подобно МДД: наличие галочки говорит о задействовании МОДа в работе сервера, а двойной щелчок вызовет окно конфигурирования.

Вернемся к настройке МДД.



По сути, *Модуль доступа к данным* – это драйвер, который, с одной стороны, реализует протокол обмена данными с заданным типом устройств, а с другой стороны, представляет эти данные в универсальном виде, пригодном для использования SCADA-системой.

Проще говоря, с одной стороны МДД обращается непосредственно к сигналам и каналам устройства, а с другой стороны представляет их в виде неких универсальных единиц – **тегов**. Теги могут быть двух типов, обусловленных двумя возможными типами сигналов: аналоговыми и дискретными. Каждый тег однозначно соответствует определенному сигналу устройства. При изменении сигнала меняется и значение тега. Запись данных тег означает подачу команды устройству на установку определенного уровня конкретного сигнала.

Для удобства восприятия теги группируются по определенным критериям, образуя **группы**.

Один МДД, как правило, позволяет обмениваться данными не с одним устройством, а сразу с несколькими, объединенными в сеть. Для того, чтобы выделить каждое устройство внутри конфигурации и дать возможность задать специфичные настройки (например, номер устройства в сети или его IP-адрес), внутри конфигурации для каждого устройства создается более крупная, чем группа, единица, которая так и называется: **Устройство**. Соответственно, единые для всех устройств настройки (номер СОМ-порта, скорость обмена, тип сети устройств и т.п.) настраиваются в свойствах **Конфигурации**.

Таким образом, иерархия представления данных в конфигурациях Модулей доступа к данным выглядит следующим образом:

- Конфигурация
- Устройство
- Группа
- Тег

После того, как МДД сконфигурирован, для него созданы устройства, группы и теги, настроены параметры связи с устройством, можно проверить обмен данными с устройствами, запустив опрос устройств. Если данные поступают, обмен происходит корректно, значит, МДД настроен правильно. Сохраняем конфигурацию, закрываем конфигуратор Сервера доступа к данным и переходим к настройке паспортов.

4. Паспорта.

Задача тегов – предоставить данные СДД в универсальном формате. Благодаря такому представлению при работе с тегами уже нет необходимости задумываться, по какому протоколу, с каких типов устройств и каким образом получены данные. Вся работу по связи с железом взял на себя МДД, это и есть его прямая задача.

Однако тег – это, как правило, сырое значение, и не обязательно он несет в себе числовое значение измеряемой величины. Это может быть код АЦП, либо величина силы тока, либо величина, выдаваемая датчиком с квадратичной шкалой измерения. В этих случаях сигнал необходимо пересчитать в реальное значение измеряемой величины.

Для такого преобразования предназначены **Паспорта** сигналов. Они на входе получают значения тегов, а на выходе выдают реальные значения величин. Помимо этого паспорта хранят в себе массу дополнительной информации: шифры, наименования сигналов, единицы измерения, способы преобразования данных и т.д. Настройка паспортов производится в **Модуле настройки паспортов**.

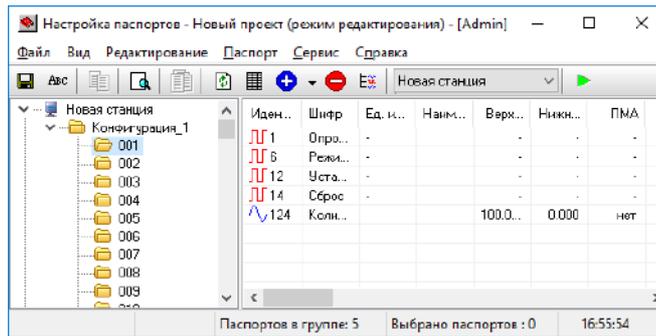


Рис.4.1. Модуль настройки паспортов.

Паспорта, как и теги, подразделяются на 2 типа сигналов: аналоговые и дискретные.

Кроме того, паспорта делятся еще и на **первичные** и **вторичные**.

Первичные паспорта непосредственно привязаны к тегам, и значение первичного паспорта напрямую зависит от значения соответствующего ему тега. Это типы паспортов **Аналоговый** и **Дискретный**. Задачей первичных паспортов является получение значений тегов и базовое преобразование этих значений: пересчет из квадратичной шкалы, масштабирование, смещение сигналов, удаление паразитных шумов и фильтрация для аналоговых сигналов, и инверсия для дискретных.

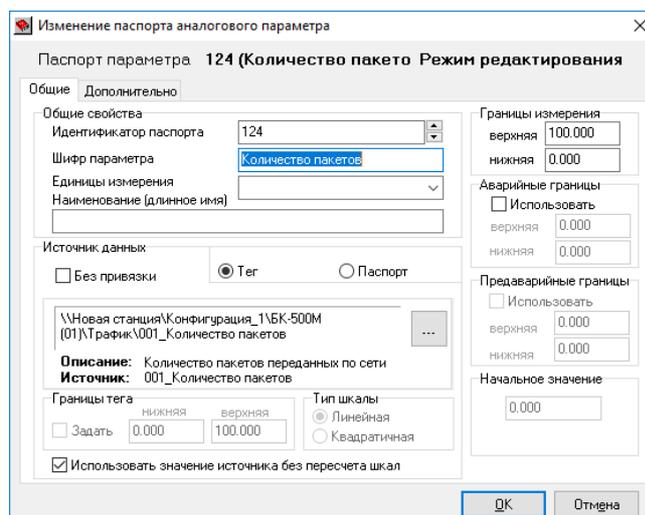


Рис.4.2. Настройка паспорта аналогового параметра

Вторичные паспорта к тегам не привязаны, источниками данных для них являются другие паспорта. Задачей вторичных паспортов является более сложная обработка данных. К вторичным относятся следующие типы паспортов: **Корректируемые**, **Сумматоры**, **Умножители**, **Формулы**, **Условия**, **Мультиплексоры**, **Дискретное управление**, **Функции** и **Скрипты**.

Самым мощным и самым универсальным типом паспорта, а поэтому и самым медленным в исполнении является паспорт-скрипт. Он позволяет производить обработку данных по алгоритмам любой сложности и ветвления, писать скрипты на C++ и Object Pascal, использовать процедуры и функции, но выполняется он относительно медленно – около 50 мс на один скрипт. Поэтому рекомендуется при использовании скриптов не создавать 2 десятка мелких скриптов, а организовать один скрипт, реализующий эти 2 десятка алгоритмов обработки.

Общий принцип настройки паспортов следующий. Создаются первичные паспорта, каждый первичный паспорт привязывается к соответствующему тегу. Для того, чтобы паспорт мог сгенерировать шифр и/или наименование из имени тега, необходимо в **Настройках** указать необходимые маски преобразования. Для того, чтобы быстро создать большое количество первичных паспортов, необходимо воспользоваться пунктом **Создать группу паспортов**.

В этом случае нам необходимо выбрать лишь теги, на основе которых будут создаваться паспорта и нажать ОК. После этого для каждого тега будет создан свой паспорт, привязанный к нему. Созданные паспорта будут сгруппированы так же, как были сгруппированы соответствующие им теги.

Точно так же, как и при настройке тегов, при настройке паспортов можно запустить опрос – тогда мы сможем наблюдать получение данных от Сервера Доступа к Данным. Очевидно, что если при опросе тегов МДД сам реализовывал связь с устройством, и при этом СДД должен был быть выгружен, то в случае с паспортами СДД наоборот, должен быть запущен. Модуль настройки паспортов в этом случае, по сути, представляет собой клиентское приложение, и запрашивает данные так же, как это делает Визуализация. И если данные модуль настройки паспортов получает корректно, то так же корректно их получит и модуль Визуализации.

Останавливаться на настройках каждого из типов паспортов не будем. Особенности настройки каждого типа достаточно подробно разъяснены в системе справки и поддержки и в руководстве пользователя.

Лекция 3. Система программирования микропроцессорных контроллеров с открытой архитектурой KLogic.

Система программирования контроллеров (СПК) KLogic является инструментом технологического программирования контроллеров в составе автоматизированных систем управления технологическими процессами, оперативно-диспетчерского управления, систем коммерческого и технического учета ресурсов.

СПК KLogic полностью интегрирована в SCADA-систему КАСКАД, но она также может использоваться отдельно от нее, с другим программным комплексом (или вообще без верхнего уровня).

1. Понятие технологической программы.

Технологическая программа содержит в себе описание окружения контроллера (дискретные и аналоговые модули ввода-вывода, устройства связи с объектами и прочая периферия), настройки связи и периодичности их опроса, описание алгоритмов работы, заданных пользователем.

Цикл выполнения технологической программы для любого контроллера выглядит следующим образом (рис. 1.1).

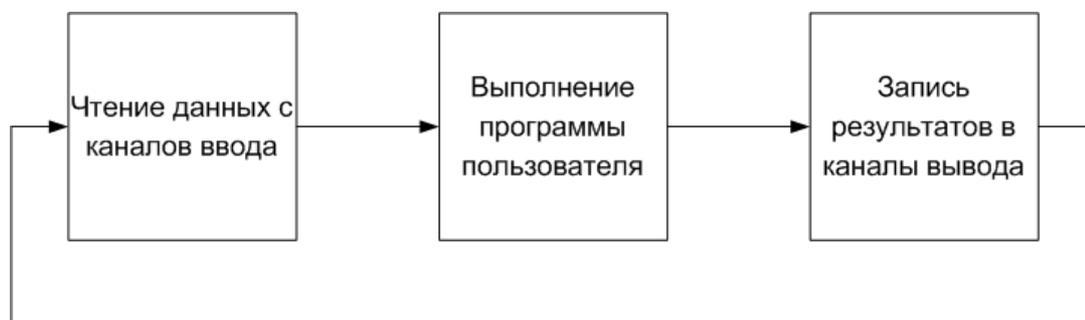


Рис. 1.1. Цикл выполнения технологической программы.

Чтение данных с каналов ввода подразумевает под собой получение текущего состояния объекта: значения необходимых технологических параметров (температура, давление, скорость) или их состояния (состояние кнопки, двигателя, выключателя). Далее происходит анализ полученного состояния объекта с использованием тех или иных инструментов, обычно под ними подразумеваются языки программирования контроллеров [МЭК 61131-3](#), либо их модификации. Вслед за проведением анализа, в контроллере происходит формирование ответной реакции на текущее состояние и его запись в каналы вывода.

2. Структура KLogic

СПК KLogic состоит из двух программных частей:

Инструментальная система, при помощи которой происходит конфигурирование и настройка контроллера. Инструментальная система (KLogic IDE) работает на платформе Win32 и предоставляет пользователю графический интерфейс, при помощи которого можно создать, загрузить и отладить технологическую программу для контроллера. Кроме набора predefined алгоритмов имеется возможность реализовывать собственные алгоритмы на двух языках программирования, максимально приближенных по синтаксису к языкам Pascal, C.

Помимо этого, при помощи инструментальной системы, можно изменить коммуникационные настройки и время контроллера, выполнить сервисные функции (удаление конфигурации, перезагрузка контроллера и прочее)

Исполнительная система, выполняющаяся на контроллере с открытой архитектурой (то есть имеющей опубликованную спецификацию работы с ним). Исполнительная система тесно работает с программными и аппаратными ресурсами контроллера, в число которых входит оперативная и

постоянная память, коммуникационные порты, встроенные каналы ввода-вывода, светодиоды. Основная задача исполнительная система - выполнение технологической программы, и дополнительные сервисные функции, в том числе коммуникации с инструментальной системой, изменение времени контроллера, конфигурации Ethernet-портов и прочее.

Исполнительная система KLogic максимально абстрагирована от конкретного оборудования и портирована на следующие целевые платформы и операционные системы:

- I7188, I8000 – MiniOS7
- МФК, ТКМ-52 – MS DOS
- Ломиконт – MS DOS
- Платформы Win32, WinCE
- Контроллер ТКМ410, операционная система eCos
- Контроллер Теконик P06, операционная система Linux
- Контроллер Деконт А9, операционная система Linux
- Контроллеры Муха UC-7112LX Plus, IA-240, операционная система Linux
- Контроллер UC DK, операционная система Linux

Взаимодействие инструментальной и исполнительной системы KLogic осуществляется по специализированному набору функций, основанных на коммуникационном протоколе Modbus, по каналам последовательной связи (COM-порт) и Ethernet. Возможность интеграции контроллеров KLogic в другие системы верхнего уровня обеспечивает поставляющийся в комплекте OPC-сервер.

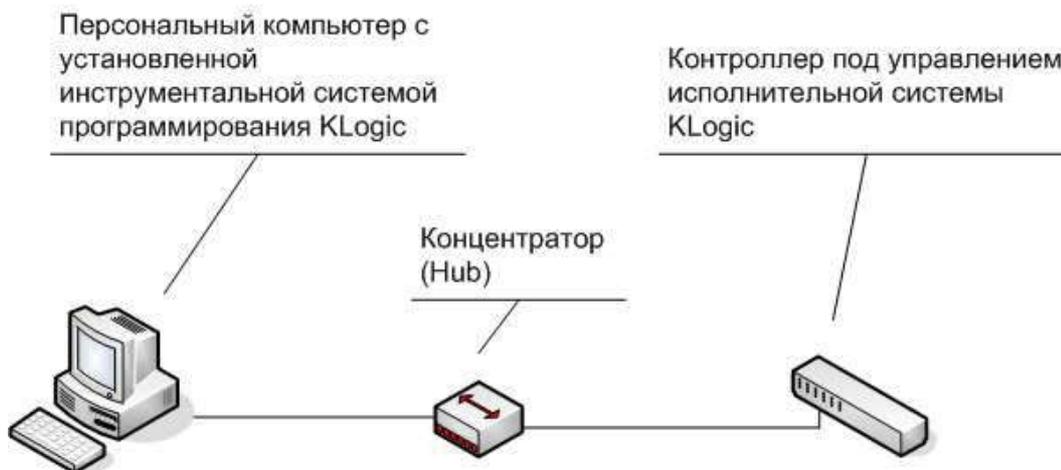


Рис. 1.2. Организация связи с контроллером.

Независимо от типа используемого контроллера и его операционной системы, все взаимодействие между компьютером и контроллером осуществляется через инструментальную систему KLogic.

Исполнительная система KLogic поддерживает сбор данных по более чем 40 протоколам. При помощи контроллера под управлением KLogic можно интегрировать различные линейки модулей ввода-вывода, счетчики различных видов ресурсов (электроэнергия, вода, тепло), терминалы релейной защиты аппаратуры и прочее оборудование

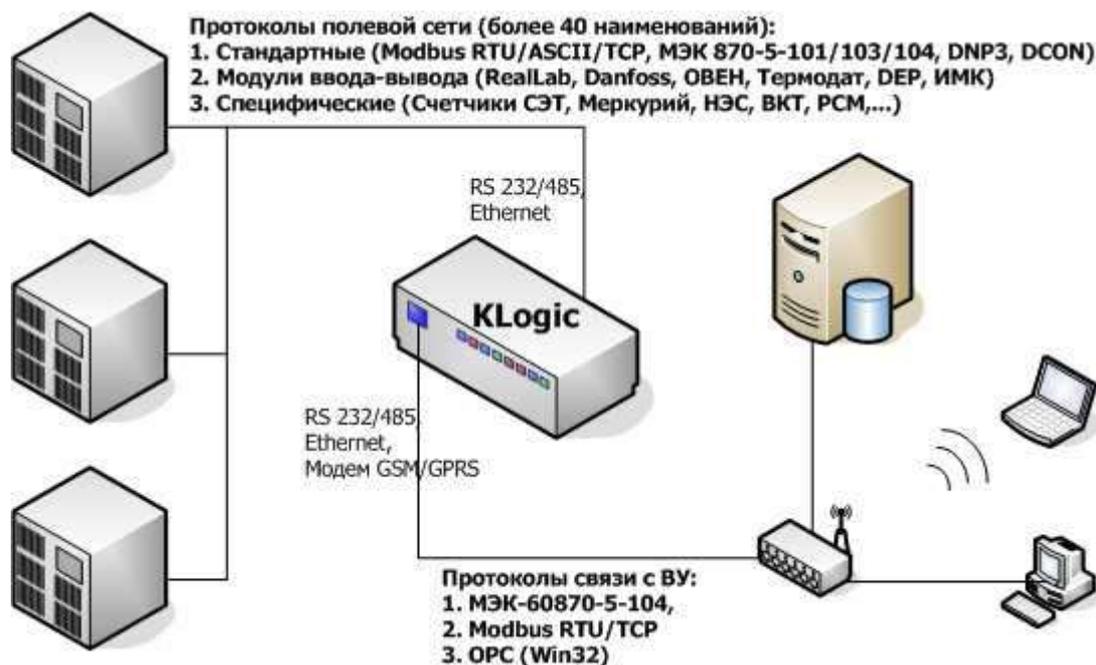


Рис. 1.3. Структура системы сбора данных и управления.

3. Архитектура исполнительной системы KLogic

Исполнительная система представляет собой один исполняемый файл. Этот файл выполняется под управлением встроенной операционной системы микропроцессорного контроллера. При запуске исполнительная система загружает конфигурацию из конфигурационного файла. На основе этой конфигурации ядро исполнительной системы запускает на выполнение ряд соответствующих задач. Конфигурационный файл генерируется инструментальной средой разработки.

Все задачи исполнительной системы KLogic в качестве входных/выходных данных оперируют параметрами, находящимися в **глобальном массиве параметров**. Доступ задач к этому массиву происходит с помощью специальных функций чтения/записи.

Глобальный массив представляет собой линейную область памяти, в которой последовательно друг за другом хранятся структуры состояния параметров. Каждый параметр однозначно идентифицируется номером.

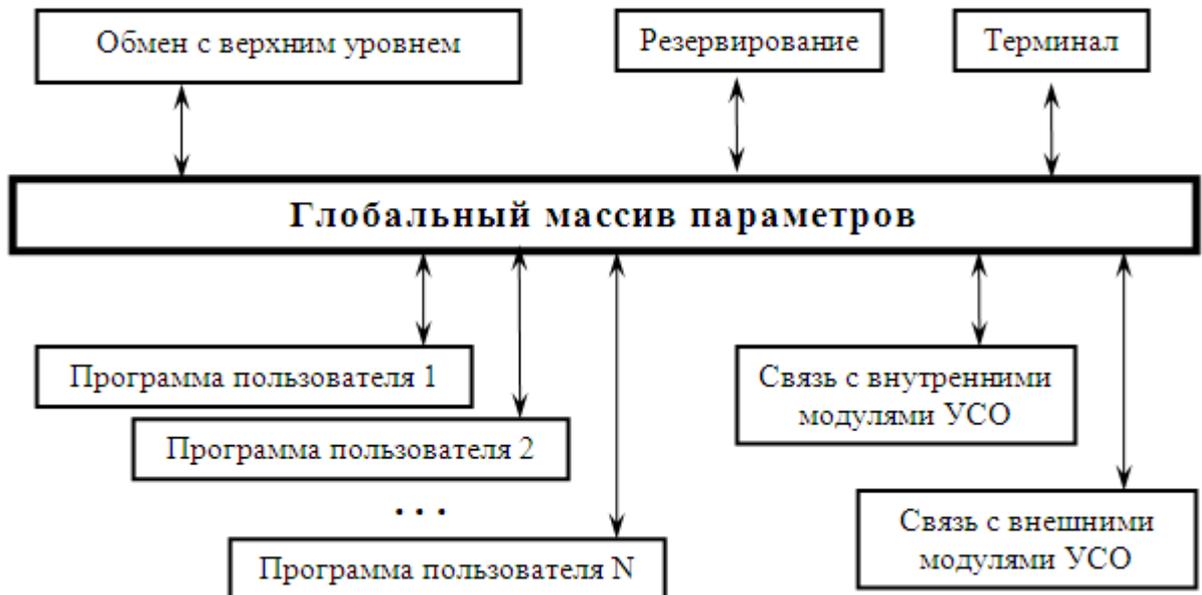


Рис. 1.4. Архитектура исполнительной системы

Все задачи в исполнительной системе контроллера работают под управлением многозадачного ядра реального времени (в DOS-подобных операционных системах), или непосредственно под операционной системой контроллера (Linux-подобные системы, Win32/WinCE). Задачи выполняются параллельно, в режиме приоритетной (вытесняющей) многозадачности. По аналогии с Windows – каждая задача представляет собой поток, выполняемый с определенным приоритетом.

Следует понимать, что задача в терминологии KLogic – это самостоятельный поток команд, выполняемый полностью параллельно, независимо от других подобных задач, и взаимодействующий с другими задачами только через массив глобальных параметров. Поэтому несколько «задач» в терминологии пользователя реально может выполняться одной задачей KLogic, если только пользователь не предпримет дополнительных усилий по распараллеливанию своих задач. Это можно делать для того, чтобы, например, выделить блоки регуляторов в отдельную задачу с более высоким приоритетом и меньшим циклом.

Приоритет	Значение	Описание
IDLE	130	Фоновый
LOWEST	120	
LOWER	110	
NORMAL	100	Нормальный
HIGHER	90	
HIGHEST	80	
REALTIME	70	Максимальный

Рис.1.5. Таблица приоритетов выполнения задач

Типы задач, выполняющихся в контроллере под управлением исполнительной системы KLogic:

- **Программа пользователя** – задача выполнения последовательности функциональных блоков, реализующая определенный алгоритм обработки параметров глобального массива. Количество таких задач и конфигурация каждой из них определяется инструментальной средой разработки.

- **Связь с внутренними модулями УСО** – задача функционирует с определенным периодом, и осуществляет чтение/запись физических контекстов ввода/вывода, находящихся непосредственно на том же микроконтроллере, на котором запущена исполнительная система. Предполагается, что эта задача будет существовать в единственном экземпляре.
- **Связь с внешними модулями УСО** – осуществляет связь с внешними модулями УСО по коммуникационным каналам связи – последовательным портам и Ethernet. Количество задач зависит от данных в конфигурационном файле.
- **Обмен с верхним уровнем** – задача обеспечивает коммуникацию верхнего уровня с исполнительной системой. Эта задача отвечает за загрузку конфигурации в контроллер, отладку программ пользователя, мониторинг состояния исполнительной системы, чтение/запись параметров глобального массива и пр. Задача существует в единственном экземпляре.
- **Резервирование** – осуществляет задачи резервирования. Существует в единственном экземпляре.
- **Терминал** – задача предоставляет средства для мониторинга состояния исполнительной системы в режиме терминала. Возможен вариант с подключением к контроллеру клавиатуры и монитора (при наличии таковых портов), и вариант удаленного терминала. В случае удаленного терминала задача использует один из последовательных портов (тот, который предусмотрен в контроллере для подобных целей).
- **Архивы** – задачи ведения оперативных и исторических архивов.
- **МЭК** – реализует поддержку обмена по протоколу МЭК 870-5-104. Контроллер является контролируемой станцией (КП).
- **Контейнер ввода-вывода** – реализует опрос разнородных модулей (разных протоколов) в пределах одной задачи ввода-вывода. Опрос может вестись по каналам связи Ethernet и последовательному порту (в том числе по модему).

Лекция 4. Клиентские модули SCADA-системы «Каскад».

1. Модуль Визуализации.

Модуль Визуализации является основным клиентским приложением SCADA-системы «Каскад». Более правильно будет назвать его «Диалоговой средой контроля и управления». Задачей этого модуля является получение данных с одного или нескольких СДД и отображение в виде набора мнемонических схем (мнемосхем), а также предоставление пользователю возможности управления. Каждая мнемосхема позволяет в схематичном и понятном пользователю виде представить участок технологической схемы с отображением необходимых параметров техпроцесса.

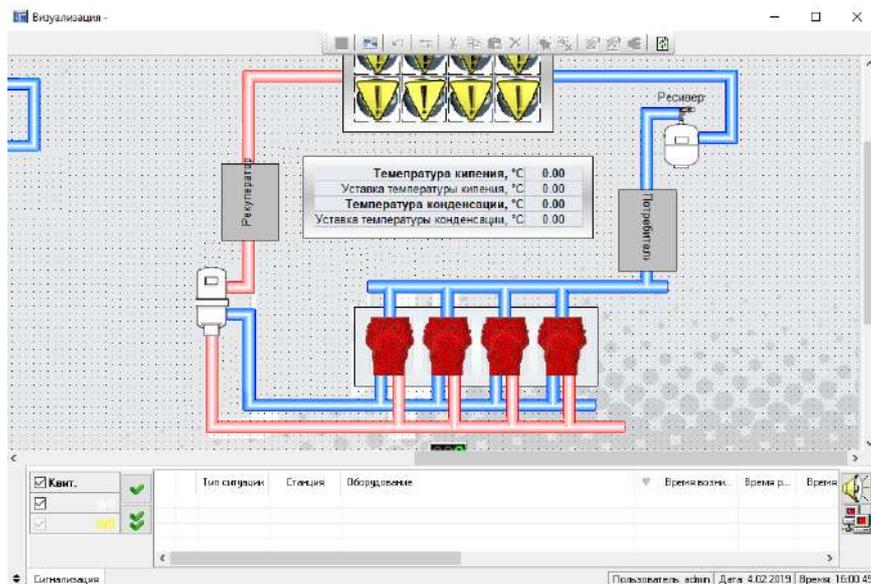


Рис.1.1. Модуль визуализации. Пример мнемосхемы.

Изображение на мнемосхеме строится при помощи объектов визуализации. Такими объектами могут быть рисунки, таблицы, текст, анимация, элементы управления, графики и т.д.

Объекты визуализации могут быть **статическими** и **динамическими**. Динамические объекты привязаны к одному или нескольким паспортам, и их отображение зависит от значений этих паспортов. Примеры таких объектов: значение параметра, гистограмма, исторический тренд.

Статические объекты к паспортам не привязываются, и их отображение всегда постоянно. На Рис.1.1. такими объектами являются трубопроводы, обозначения датчиков (РТ, ФЕ, ТЕ), надписи.

Для перехода по мнемосхемам служат объекты: кнопки перехода, панель навигации (на рис.1.1 она сверху) с соответствующим набором кнопок, а также дерево проекта и кнопки клавиатуры PgUp и PgDown.

Для внесения изменений в проект (создания, удаления и настройки мнемосхем) необходимо перейти в режим редактирования, нажав клавишу **F6** (рис. 1.2).

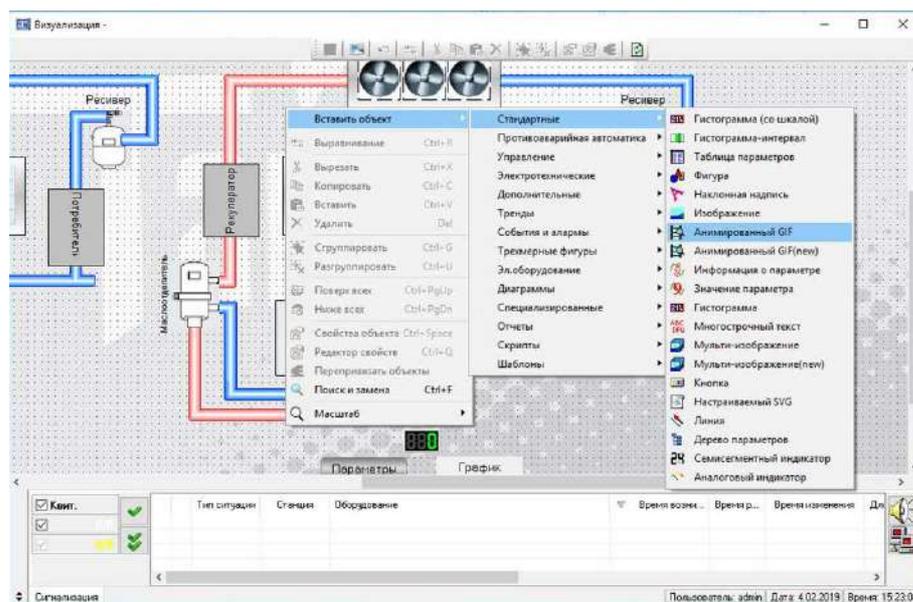


Рис. 1.2. Визуализация. Режим настройки.

Объекты визуализации сгруппированы по смыслу и функциональному назначению. После добавления на мнемосхему объекты можно двигать, изменять их размер при помощи мыши, копировать, удалять.

Каждый объект имеет настраиваемые свойства. Настроить объект, можно щелкнув по нему дважды левой кнопкой мыши, либо вызвав на нем контекстное меню и выбрав пункт **Свойства объекта**.

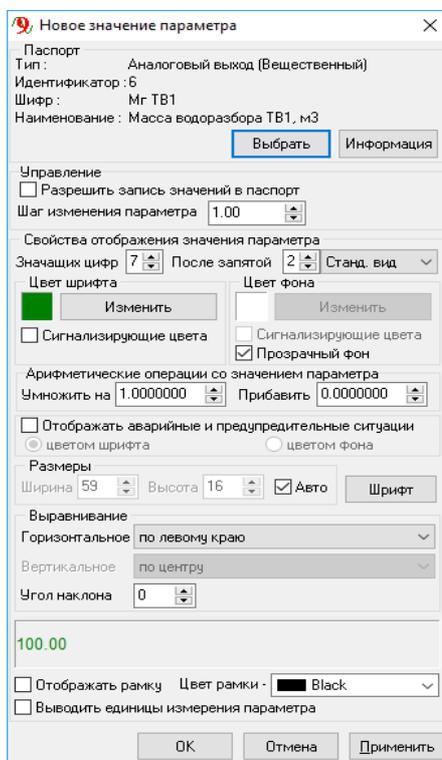


Рис.1.3. Свойства объекта «Значение параметра».

В случае, если необходимо установить один и тот же параметр нескольких одинаковых или однотипных объектов в одно и то же значение, удобнее воспользоваться **Редактором свойств** (он вызывается из контекстного меню либо нажатием комбинации <Ctrl+Q>) предварительно выделив нужные объекты (рис. 1.4).

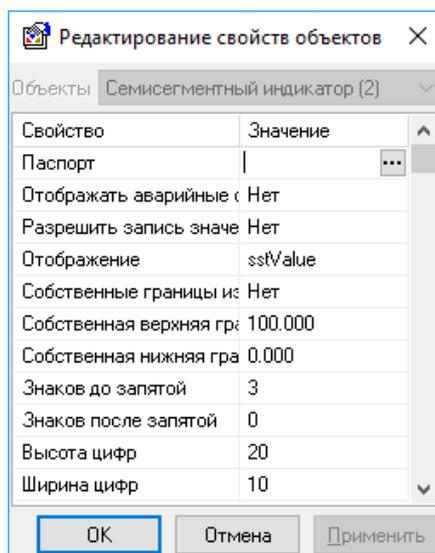


Рис. 1.4. Групповое редактирование свойств объектов.

Помимо этого объекты, связанные по смыслу, можно сгруппировать. Для этого нужно выделить их и нажать <Ctrl+G>. После этого с объединенными в группу объектами можно работать как с одним объектом: перемещать, копировать, масштабировать и т.д.

У групп есть еще одно интересное свойство. Можно управлять видимостью группы, то есть при необходимости скрывать ее или отображать. Для этого необходимо привязать группу к дискретному паспорту, и в зависимости от его состояния группа будет либо видна, либо нет. Привязку можно выполнить так: в дереве проекта (вызвать его можно по F3) на вкладке *Объекты* нужно найти нужную группу, щелкнуть на ней правой кнопкой мыши и выбрать **Свойства**. Появится окно настройки группы (рис.1.5).

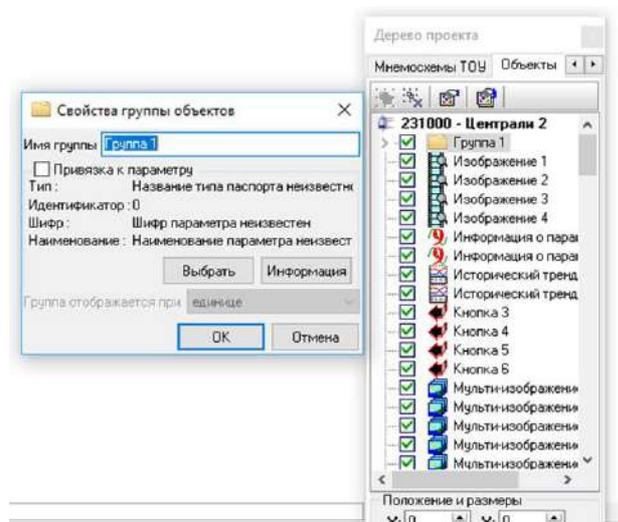


Рис. 1.5. Свойства группы объектов.

Как и Сервер Доступа к Данным, Визуализация построена по модульному принципу. Есть единая оболочка, имеющая необходимые интерфейсы для создания, удаления и обслуживания объектов, и есть набор библиотек, написанных с соблюдением определенных правил, и выполняющих различные функции отображения. Каждая библиотека может хранить один или несколько объектов визуализации. При необходимости пользователь может сам написать объект с нужными свойствами и подключить его к системе.

Как правило, объекты, из которых состоит мнемосхема, загружаются вместе с самой мнемосхемой. Вместе с ней они создаются и вместе с ней уничтожаются. При переключении на следующую мнемосхему предыдущая уничтожается не сразу. Модуль Визуализации хранит в памяти кэш из несколько последних мнемосхем (по умолчанию 5). И только когда этот кэш заполняется, наиболее старые мнемосхемы удаляются из него. Это связано с тем, что, как правило, в работе используются несколько одних и тех же мнемосхем, которые отображают сводную информацию о техпроцессе, и оператор в штатном режиме переключается только между ними. Сохраняя в памяти их в памяти, мы ускоряем переключение между ними, поскольку при этом не требуется каждый раз создавать мнемосхему, все находящиеся на ней объекты и запускать процессы опроса параметров.

Однако когда мнемосхема переходит в кэш, опрос параметров с нее прекращается, чтобы не нагружать протокол опроса и не запрашивать ненужную информацию. Если все же необходимо, чтобы мнемосхема всегда получала данные, даже находясь в кэше, необходимо в ее настройках включить

2. Модуль регистрации технологических параметров.

Одной из основных функций любой SCADA-системы является сохранение истории изменения значений технологических параметров в базы данных. В SCADA-системе «КАСКАД» за выполнение этой функции отвечает подсистема регистрации технологических параметров, основными частями которой являются:

- 1) модуль настройки баз данных (БД) технологических параметров (ТП);
- 2) модуль регистрации значений ТП в БД;
- 3) модуль извлечения данных из БД ТП.

2.1. Модуль настройки БД ТП

Данный модуль предназначен для создания, настройки, удаления и других действий с базами данных. Количество БД ТП в проекте – не ограничено.

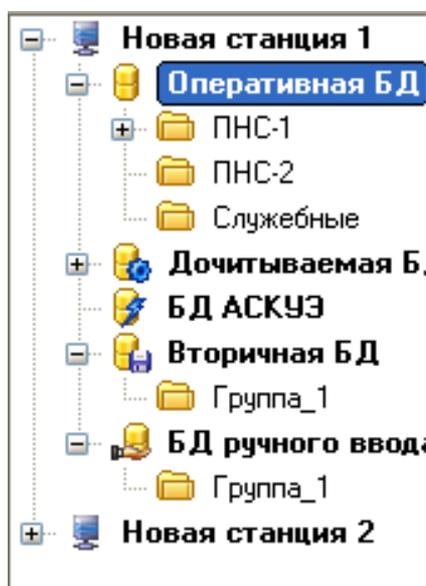


Рис. 2.1. Список БД ТП

Существует несколько типов БД ТП:

- 1) *Обычная (или оперативная) БД* – база данных регистрации оперативных значений. В неё записываются текущие значения параметров. Это наиболее часто используемый тип БД.
- 2) *БД с дочиткой* – база данных регистрации исторических значений. В такую базу записывается информация, получаемая из архивов устройств сбора данных (данный тип БД может использоваться при условии, что контроллеры имеют собственные исторические архивы и в модуле доступа к данным для этих контроллеров реализовано чтение исторических значений).
- 3) *БД АСКУЭ* – база данных регистрации параметров Диспетчер-АСКУЭ (рассмотрение данного типа БД выходит за рамки нашего курса).
- 4) *Вторичная БД* – база данных с обработкой. В неё записываются значения, периодически получаемые из первичных БД (*Обычных БД, БД с дочиткой, БД ручного ввода*) и обработанные по заданному алгоритму: среднее, сумма, минимум, максимум, смещение, масштабирование и т.д.
- 5) *БД ручного ввода* – база данных, информация в которую заносится пользователем вручную, с помощью объекта визуализации «Ручной ввод».

Каждая БД ТП имеет свои настройки, в зависимости от типа. Тем не менее, существует ряд свойств, присущих большинству БД:

1. Для каждой БД ТП обязательно необходимо задать параметры подключения к файлу БД и создать собственно файл БД.
2. Ограничение срока хранения данных – признак, отвечающий за автоматическое удаление устаревших данных из БД.
3. Список параметров – набор параметров, данные по которым будут сохраняться в БД. Параметры могут объединяться в группы неограниченной вложенности. Добавление параметров в БД осуществляется простым перетаскиванием паспортов или их групп. Список параметров БД определяется инженером на основе требований проекта. В базу данных следует добавлять только те параметры, история которых представляет интерес.
4. Настройки записи определяют, по каким алгоритмам будет происходить запись значений параметров в базу данных:
 - *периодически;*
 - *по расписанию;*
 - *по событию* (при срабатывании заданного дискретного параметра);
 - *по инициативе снизу* (при получении уведомления от нижнего уровня SCADA-системы о том, что значение параметра изменилось).

Кроме этого, для БД ТП можно настроить условия записи, резервирование, автоматическое создание архивных копий, а также посмотреть статистику по БД или очистить базу данных от устаревших значений.

По умолчанию, группы параметров имеют те же самые настройки, что и БД. При необходимости, для каждой группы можно задать свои собственные настройки, отличающиеся от настроек корневого элемента (БД или родительской группы параметров).

В подсистеме регистрации технологических параметров существует важное правило: для каждого паспорта можно создать только один параметр БД ТП. Данное ограничение позволяет однозначно определить параметр БД на основе выбранного паспорта, что используется, например, в трендах и рапортах.

2.2. Модуль регистрации значений ТП в БД

Данный модуль является одним из модулей обработки данных (МОД), входящих в сервер доступа к данным (СДД). Он отвечает за сохранение значений параметров в базы данных на основе заданных настроек.

2.3. Модуль извлечения данных из БД ТП

Это библиотека с открытым интерфейсом для доступа к информации, хранящейся в БД, из клиентских приложений. Данный модуль позволяет осуществлять просмотр накопленных данных на трендах и формировать отчёты.

3. Модуль формирования рапортов.

SCADA-система «КАСКАД» непрерывно ведёт историю технологического процесса. Для того, чтобы проанализировать накопленную информацию, создать некий сводный отчет за определенный период (смену, сутки, месяц), служит подсистема создания отчётной документации.

Создание отчётной документации в SCADA-системе «КАСКАД» осуществляется с помощью **модуля формирования рапортов**, основными функциями которого являются:

- 1) создание и настройка шаблонов отчётов;
- 2) генерация отчётов на основе настроенных шаблонов;
- 3) отображение готовых отчётов, с возможностью их печати, сохранения и экспорта в Microsoft Excel, HTML или текстовый файлы.

Модуль формирования рапортов имеет два режима: рабочий режим и режим настройки. Рабочий режим служит для навигации по списку настроенных шаблонов рапортов, генерации отчётов, их просмотра, печати и сохранения. В режиме настройки, как следует из названия, осуществляется создание и вся настройка необходимых отчётов.

Для успешной настройки шаблонов отчётов следует понимать назначение трёх основных элементов модуля формирования рапортов:

- 1) дерево рапортов;
- 2) дерево алгоритмов;
- 3) таблица шаблона.

1) **Дерево рапортов** (рис. 3.1) позволяет переключаться между существующими шаблонами рапортов, добавлять новые шаблоны, разбивать их по группам, удалять ненужные, создавать шаблоны на основе уже существующих с помощью копирования и вставки, настраивать основные свойства рапортов.

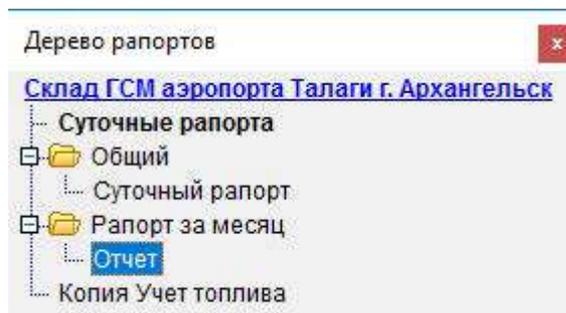


Рис. 3.1. Дерево рапортов

2) **Дерево алгоритмов** (рис. 3.2) позволяет настраивать набор алгоритмов, содержащихся в рапорте. Данный набор определяется инженером на основе требований, которым должен удовлетворять тот или иной отчёт. К примеру, если рапорт должен содержать информацию по заданным параметрам за заданный промежуток времени, в виде таблицы с заданным шагом, то следует выбрать алгоритм «Таблица значений». Другой пример: если необходимо настроить отчёт, который бы выдавал список событий за заданный промежуток времени с заданными критериями отбора, то следует выбрать алгоритм «Список событий». Таким образом, в каждом конкретном случае инженер подбирает набор алгоритмов, необходимых для конкретного отчёта. Модуль формирования рапортов предоставляет большой выбор алгоритмов, достаточный для того, чтобы настроить отчёт любой сложности.

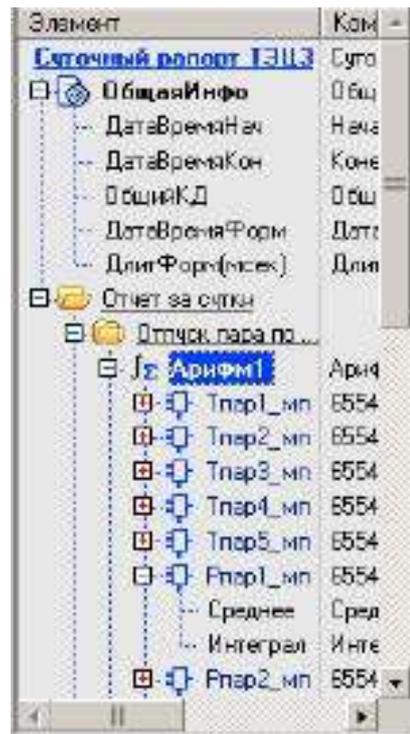


Рис. 3.2. Дерево алгоритмов

С помощью дерева алгоритмов осуществляется добавление алгоритмов в шаблон отчёта, их группировка, настройка свойств каждого алгоритма, размножение алгоритмов с помощью механизма копирования и вставки.

Каждый алгоритм решает определённую задачу и имеет свой набор настроек. Тем не менее, существует ряд общих настроек, присущих большинству алгоритмов, как например:

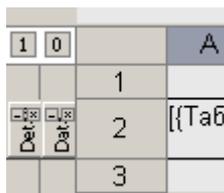
- интервал времени, за который необходимо запросить данные для отчёта;
- список параметров, данные по которым необходимо отобразить в отчёте.

Алгоритмы располагаются в виде дерева, корневым элементом которого является имя рапорта. Они могут объединяться в группы неограниченной вложенности. При этом каждый алгоритм имеет признак наследования интервала времени от родительской ветки, что позволяет наследовать родительский диапазон или задавать свой собственный.

Каждый алгоритм также имеет свой набор выходов, в зависимости от решаемой задачи. Выходы алгоритмов соответствуют конкретным значениям, которые будут отображаться в сгенерированных отчётах. Примеры выходов: сумма, среднее, минимальное значение, шифр параметра, значение параметра, временная метка параметра, номер строки в таблице, зона производства, текст события, имя пользователя и т.д.

Выходы алгоритмов расставляются в ячейки таблицы шаблона с помощью простого перетаскивания мышью.

Выходами алгоритмов могут быть как одиночные значения, так и наборы данных. Во втором случае в таблице шаблона, при перетаскивании выхода в ячейку, автоматически создаются бэнды (рис. 3.3) – специальные элементы, которые привязаны к своему набору данных и размножаются в готовом отчёте по числу записей в этом наборе. Например, для алгоритма «Таблица значений», настроенного на сутки с периодом в один час, в таблице шаблона будет соответствовать всего одна строка с автоматически созданными бэндами, которые размножат эту строку в готовом отчёте до 24 строк.



1	0		A
		1	
		2	{Tab.}
		3	

Рис. 3.3. Бэнды

3) **Таблица шаблона** (рис. 3.4) – это excel-подобная таблица с ячейками, с помощью которой настраивается внешний вид отчёта. Ключевыми элементами таблицы являются выходы алгоритмов, которые в готовом рапорте заменяются на реальные данные.

Таблица шаблона предоставляет широкие возможности для оформления отчёта: шрифты, цвет, выравнивание, объединения, форматирование значений, заливка, штриховка, добавление листов, параметры страницы для печати, настройка колонтитулов и т.д.

ТЭЦ-3							
A	B	C	D	E	F	G	
10			о том, что отпуск тепловой энергии от источника теплоты э				
11			с [ОбщаяИнфо.ДатаЕ по [ОбщаяИнфо.ДатаВрем				
12			на основании базы данных параметров				
13							
14			А. ПО ПАРОВЫМ МАГИСТРАЛЯМ				
15			1. Отпуск пара				
16							
17			Источник пара	Номер паропровода	P, МПа (кгс/см2)	t, °C	h, кДж/кг (ккал/кг)
18							
19							
20			ТЭЦ-3	1 (Химпром)	[Арифм1.П.П655435.Среднее]	[А	
21			ТЭЦ-3	2 (Химпром)	[Арифм1.П.П655436.Среднее]	[А	
22			ТЭЦ-3	3 (Химпром)	[Арифм1.П.П655437.Среднее]	[А	
23			ТЭЦ-3	4 (Химпром)	[Арифм1.П.П655438.Среднее]	[А	
24			ТЭЦ-3	5 (Химпром)	[Арифм1.П.П655439.Среднее]	[А	
25					Всего		
26			ТЭЦ-3	6 (НГТС)	[Арифм2.П.П655368.Среднее]	[А	
27			ТЭЦ-3	7 (резерв)	[Арифм2.П.П655373.Среднее]	[А	
28			ТЭЦ-3	8 (резерв)	[Арифм2.П.П655378.Среднее]	[А	

Рис. 3.4. Таблица шаблона

Отдельного упоминания заслуживают формулы, которые можно использовать в таблице шаблона в случае необходимости. Элементами формул могут являться как ячейки, так и выходы алгоритмов. Формулы записываются в ячейках, начиная со знака равенства, и могут содержать следующие операции и функции: арифметические, логические, математические, статистические, текстовые и т.д. Полный перечень формул с примерами можно изучить в справке для модуля формирования рапортов.

С помощью возможностей таблицы шаблона, а также формул, можно настроить отчёт любой сложности. В редких случаях, когда не хватает функциональности имеющихся механизмов, можно использовать скрипты. По умолчанию, скрипт содержит заготовку для процедуры, которая будет выполняться после генерации отчёта. Такая процедура может понадобиться в том случае, когда необходимо автоматически вносить изменения в готовую таблицу уже после генерации рапорта.

В заключение следует отметить, что модуль формирования рапортов является СОМ-сервером, предоставляющим ряд методов для формирования отчётов, их просмотра, печати и т.п. Эта особенность позволяет, во-первых, осуществлять генерацию рапортов с мнемосхем модуля визуализации, а во-вторых, организовать автоматизированное формирование рапортов по заданному расписанию.

Для автоматизированного формирования рапортов по расписанию необходимо создать скрипт (текстовый файл), содержащий набор команд на основе методов, предоставляемых СОМ-сервером. Список имеющихся методов с примерами, можно изучить в справке для модуля формирования рапортов.

4. Модуль просмотра истории.

Как мы уже рассмотрели, модуль Визуализации позволяет оперативно просматривать информацию о технологическом процессе и управлять им. В том числе визуализация позволяет просмотреть графики – как за ранее созданные наборы, так и динамически выбираемые при помощи специального инструмента – быстрого тренда.

Однако на практике нередко возникает задача динамического анализа поступивших данных в графическом виде. При этом хотелось бы опираться на предварительно созданные наборы графиков, но и иметь возможность «на лету» добавить или удалить один или несколько графиков, соотнести несколько наборов графиков одновременно и т.д.

Для такого анализа хорошо подходит следующий модуль SCADA-системы «Каскад» - это **модуль просмотра истории**.

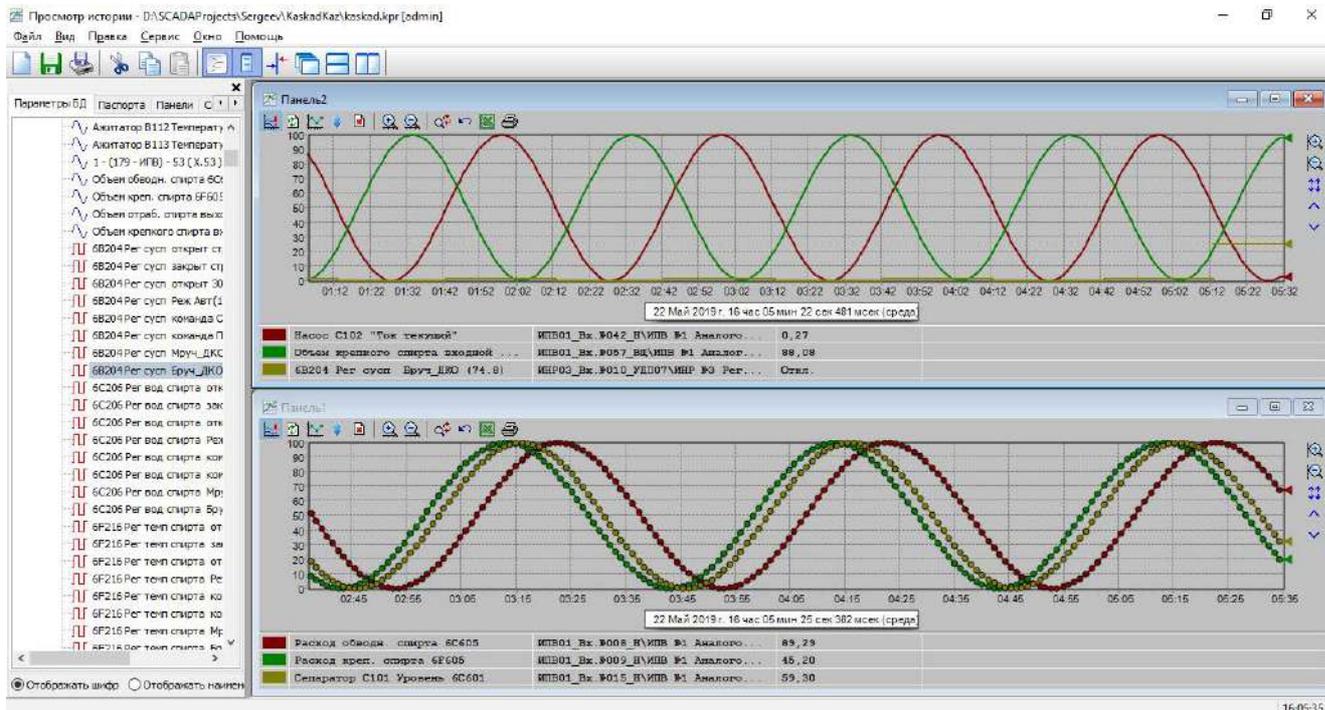


Рис.3.1. Модуль просмотра истории.

Задачей модуля просмотра истории является выдача накопленных исторических данных в графическом виде. Все графики организуются в наборы – **панели предыстории**. Количество графиков на каждой панели в принципе не ограничено, но особого смысла в отображении большого количества графиков нет, поскольку происходит загромождение панели, и, как следствие, снижение удобства анализа информации.

В левой части каждой панели выводится список графиков, значение каждого графика в момент времени под курсором и цвет графика. В правой части панели находятся сами графики.

На рис. 3.1. обе панели находятся в *следающем режиме*, когда панели периодически обращаются к базам данных и дочитывают информацию по каждому графику на текущий момент. Панель предыстории при этом плавно передвигается по оси времени так, чтобы курсор всегда указывал на текущее время, то есть *следа за текущим моментом*.

Для того, чтобы уменьшить или увеличить интервал времени, за который выводятся графики, можно воспользоваться кнопками *Left* и *Right* на дополнительной клавиатуре. Точно так же для изменения масштаба по вертикали необходимо воспользоваться кнопками *Up* и *Down* на дополнительной клавиатуре.

Для того, чтобы иметь возможность перемещаться по оси времени, необходимо выйти из следающего режима (*F7*). После этого для перемещения можно использовать кнопки навигации основной клавиатуры: *Влево*, *Вправо*, *Вверх* и *Вниз*.

Каждый график можно настроить дополнительно: задать ему толщину линии, цвет, ступенчатость, стиль линии и т.д. (рис. 3.2).

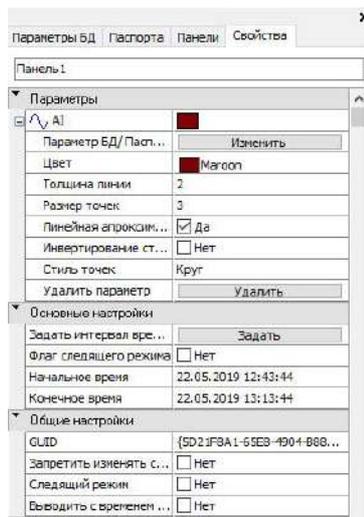


Рис. 3.2. Настройки графика.

После того, как панель настроили, ее можно сохранить для последующего использования. Если есть необходимость сохранения настроенной группы панелей – есть и такая возможность. Для этого нужно сохранить **Рабочий стол**. Тогда все открытые панели будут сохранены с учетом своих позиций.

Для анализа процесса часто требуется выровнять все панели так, чтобы они отображали данные за один и тот же период времени, и чтобы курсор каждой панели указывал так же на один и тот же момент времени. Такое выравнивание осуществляется при помощи команды **Синхронизировать все панели по текущей**, или нажатием **F3**.

Есть в модуле просмотра истории и еще один удобный инструмент. Выйдите из следящего режима, нажмите и удерживайте клавишу **<Ctrl>**, щелкните левой кнопкой мыши на панели и, не отпуская, ее, выделите некоторый интервал (рис. 3.3).

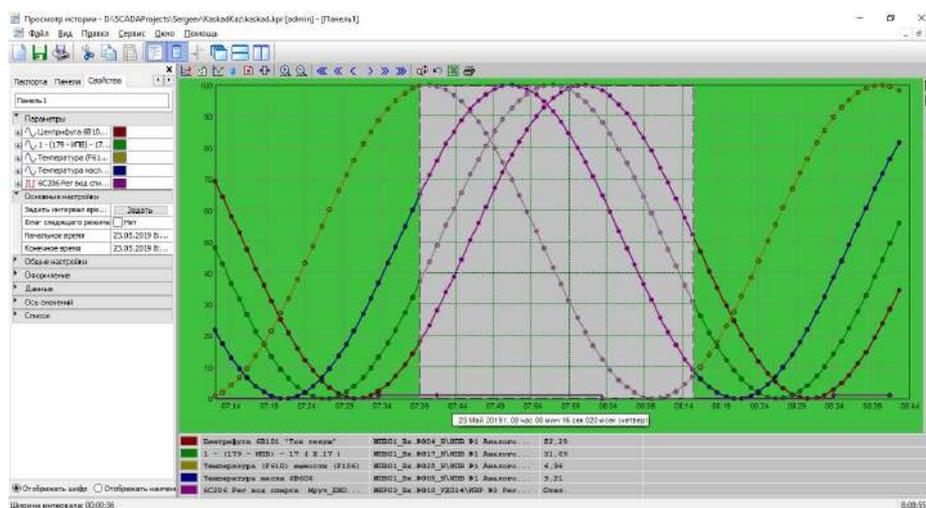


Рис. 3.3. Выделение интервала.

Всплывающая подсказка покажет ширину выделенного интервала, а если нажать клавишу **Пробел**, выделенный интервал будет развернут на всю панель.

Когда пользователь настроил панель предыстории под себя (задал цвет фона, шрифты и т.д.), ему, обычно, желательно, чтобы все последующие панели создавались с такими же настройками. Такая возможность в модуле просмотра истории есть. Для того, чтобы сохранить текущую панель в качестве шаблона, необходимо нажать в окне настройки кнопку **Установить по умолчанию**.

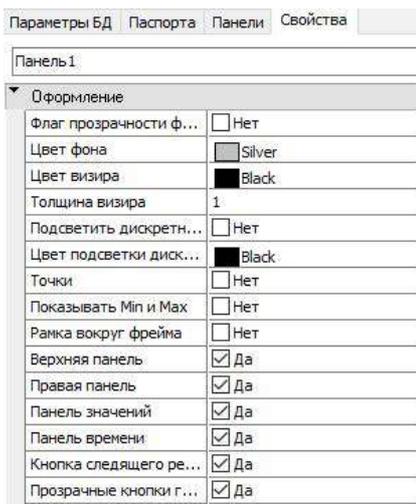


Рис. 3.4. Настройка панели.

Помимо всего перечисленного, модуль просмотра истории предоставляет множество дополнительных функций: можно настроить внешний вид каждой панели, указать и раскрасить заголовки, распечатать панели, экспортировать точки в формат Microsoft Excel и т.д. Подробнее со всеми этими возможностями можно самостоятельно при помощи системы справки и поддержки.

И еще один маленький нюанс, который, возможно, пригодится при настройке проектов.

Как правило, настроенный проект автоматизации не хранится у каждого клиента, особенно если таких клиентов много. Он выкладывается администратором в сеть на определенном сетевом ресурсе, а пользователям выводятся лишь ярлыки. Сам ресурс, как правило, доступен только по чтению. Поэтому пользователь, настроив свои панели предыстории и рабочие столы, сохранить свои настройки на сервер не сможет. Чтобы дать возможность пользователю сохранить свои настройки в нужное ему место, в модуле просмотра истории предусмотрена возможность изменения папки хранения личных настроек. Задать расположение этой папки можно при помощи пункта **Файл->Папка хранения настроек**.

Лекция 5. Модуль аварийной и предупредительной сигнализации.

1. Общие сведения.

Задачей модуля аварийно-предупредительной сигнализации (АПС) является контроль за состоянием параметров технологического процесса и уведомление персонала обо всех случаях его нарушения. Кроме того, уведомление может происходить и о событиях, не являющихся авариями, например, об изменении режима работы оборудования, уставок, либо, например, о проникновении посторонних в помещение.

В случае возникновения ситуации, о которой необходимо оповестить персонал, модулем АПС генерируется *аларм*.

Аларм – это сообщение, генерируемое модулем аварийной и предупредительной сигнализации, предназначенное для информирования пользователей системы о выходе параметра за допустимые границы или о переходе системы в недопустимый режим.

Оповещение может осуществляться следующими способами:

- Звуковым сигналом
- Речевым сообщением
- Коротким сообщением sms
- Сообщением электронной почты (e-mail)
- Записью в БД событий
- Визуальной индикацией
- Подсветкой соответствующих объектов и кнопок перехода.

При генерировании аларма делается запись в базе данных событий с подробным указанием данных о том, что за ситуация сработала, для какого параметра, в какое время. Очень важно в дальнейшем при анализе аварийной ситуации знать хронологию возникновения событий и действий персонала. Поэтому в записи для аларма в БД событий фиксируется следующая крайне важная информация:

- Время перехода параметра в аварийное (сигнализируемое) состояние;
- Время уведомления пользователя об аварии, то есть время генерации аларма;
- Время квитирования аларма пользователем, то есть время, когда пользователь воспринял информацию об аварии и начал принимать меры по ее устранению;
- Время выхода параметра из сигнализируемого состояния;
- Отметка о квитировании (если параметр был сквитирован);
- Имя пользователя, сквитировавшего аларм;
- АРМ, с которого было произведено квитирование.

Поясню значение термина *квитирование*.

Квитирование – это уведомление пользователем системы аварийно-предупредительной сигнализации о том, что он воспринял информацию об аварии и приступил к принятию необходимых мер по ее устранению.

Аларм может быть активным и квитированным. Только что сгенерированный аларм всегда активен.

Информация о сработавших алармах передается на клиентские приложения, подключенные к серверу АПС. Как правило, это модуль визуализации. Модуль визуализации в соответствии со своими настройками осуществляет звуковое и визуальное оповещение пользователя об алармах, а также дает пользователю возможность сквитировать звучащие алармы с записью в БД событий времени квитирования, имени сквитировавшего пользователя и ip-адреса АРМа, с которого было произведено квитирование (если, конечно, пользователю было разрешено квитирование в модуле настройки прав пользователей). Для этого в модуле Визуализации предусмотрен соответствующий модуль расширения. В случае, если объект визуализации отображает аварийные ситуации, при наличии несквитированного аларма вокруг объекта мигает рамка с цветом, соответствующим типу ситуации (авария либо предупреждение). Если аларм квитирован, рамка горит постоянно. Каждый клиент способен получать информацию одновременно с нескольких серверов сигнализации.

2. Настройка модуля АПС.

Принцип настройки модуля аварийной и предупредительной сигнализации схож с настройкой модуля регистрации баз данных технологических параметров.

Визуально модуль состоит из трех частей: дерева паспортов, дерево групп сигнализации, в центре – окно настройки групп.

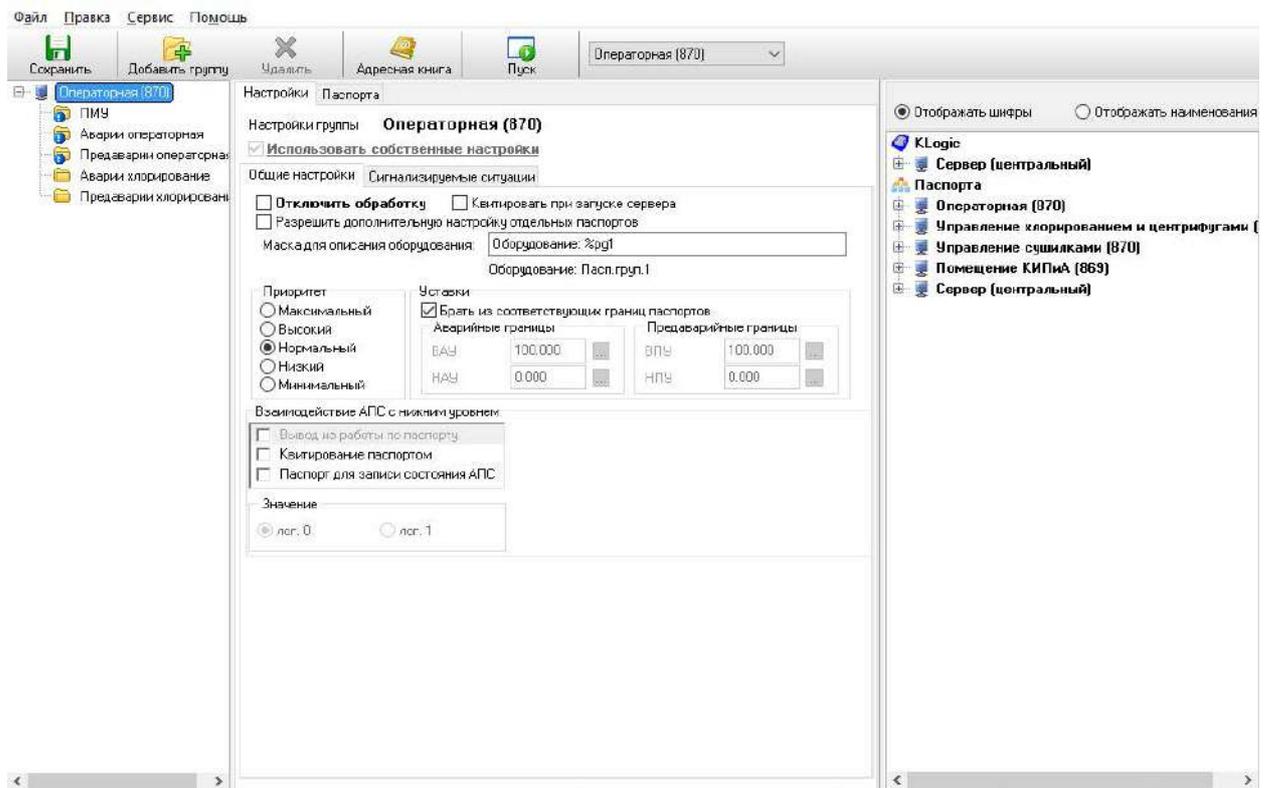


Рис. 1. Конфигуратор модуля АПС.

Для простоты настройки алармы должны группироваться по логическим признакам срабатывания: параметры, имеющие сходные настройки сигнализации, желательно помещать в одну группу сигнализации.

В таком случае настройка сигнализации в проекте сводится к правильной настройке группы и помещению в нее паспортов нужных параметров.

Рассмотрим настройку групп сигнализации.

Для каждой группы можно задать приоритет генерируемых ею алармов: чем выше приоритет аларма, тем раньше он будет просигнализован пользователю. Если несколько алармов имеют одинаковый приоритет, их проигрывание будет осуществляться последовательно.

Для аналоговых параметров, входящих в группу, можно принудительно задать аварийные и предаварийные границы (уставки), либо взять их из настроек паспортов. Значения этих границ определяют аварийные уставки:

- ВАУ – верхняя аварийная уставка;
- ВПУ - верхняя предаварийная уставка;
- НПУ - нижняя предаварийная уставка;
- НАУ – нижняя аварийная уставка.

Для дискретных параметров, входящих в группу, эта настройка будет проигнорирована.

Непосредственно за генерацию алармов отвечают сигнализируемые ситуации. Для каждой группы могут быть настроены следующие типы сигнализируемых ситуаций:

- ВАУ, НАУ, НПУ, НАУ
- Включение/выключение/переключение параметра
- Нахождение параметра выше или ниже заданных границ, либо нахождение параметра в заданном диапазоне.
- Наличие определенного качества значения параметра.

Для каждой аварийной ситуации указывается ее тип: авария, предупреждение, сообщение.

Авария – это критическое нарушение хода технологического процесса, ставящее под угрозу качество выпускаемого продукта, либо могущее привести к выходу из строя оборудования, либо несущее угрозу жизни и здоровью людей, либо угрожающее окружающей среде, - требующее немедленного вмешательства технологического персонала.

Аварии имеют наивысший приоритет, и информирование о них происходит в первую очередь.

Предупреждение – это такое состояние параметра технологического процесса, в котором он еще находится в диапазоне допустимых значений, но приближается к зоне аварийных значений, либо нарушение технологического процесса носит не критический характер, однако требует вмешательства технологического персонала.

Сообщение – это уведомление о нормальном ходе технологического процесса. Это может быть либо уведомление об изменении режима работы, либо штатное изменение хода технологического процесса. Звуковой и визуальной сигнализации в этом случае нет, реакции пользователя не требуется. Событие фиксируется только в базе данных событий.

Для каждой ситуации независимо от других можно настроить следующие параметры:

- Запись сообщений в БД событий
- Отправку сообщений смс и электронной почты и тексты этих сообщений
- Настройку проигрываемых звуковых файлов
- Время подтверждения ситуации
- Необходимость зашелкивания ситуации.

Два последних пункта, думаю, требуют пояснения.

Время подтверждения ситуации позволяет исключить ложные либо кратковременные срабатывания, не представляющие интереса для оператора.

На рис. 2 это время обозначено как $t_{\text{дост}}$.

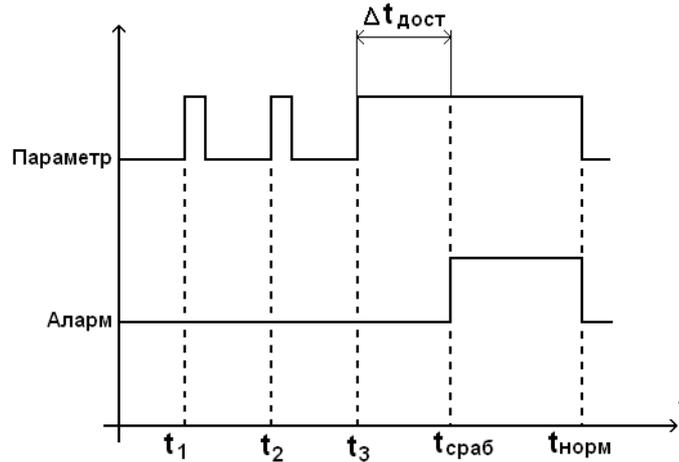


Рис. 2 . Время подтверждения ситуации.

Защелкивание ситуации решает прямо противоположную задачу. В случае, контроля особенно важных параметров, когда даже кратковременное нарушение имеет большое значение, есть риск пропустить аларм, если ситуация сработала очень кратковременно. В этом случае срабатывание должно быть зафиксировано, и звуковая сигнализация должна звучать до тех пор, пока оператор не произведет квитирование аларма, либо пока не сработает более приоритетная ситуация.

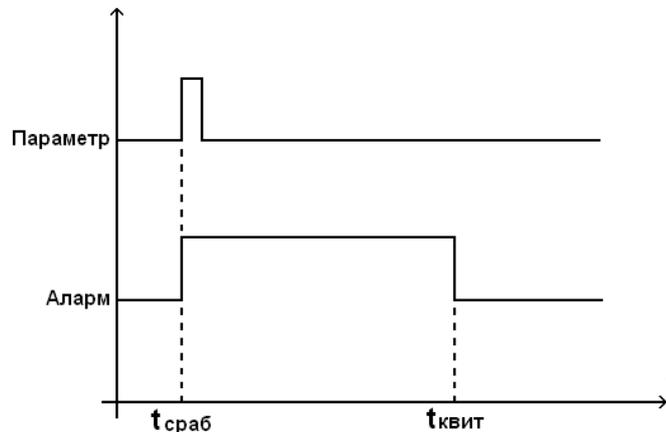


Рис. 3. Защелкивание ситуации.

В случае, если аларм с защелкой сработал, но не был сквитирован пользователем до того, как параметр успел вернуться в зону нормальных значений, а затем сработал повторно – в этом случае предыдущая запись об аларме квитируется с указанием в поле имени пользователя «повторное срабатывание», и тут же генерируется новый аларм с новой временной меткой.

Несмотря на то, что параметры желательно группировать по логике обработки и приоритетам, вполне возможна ситуация, когда для различных алармов необходимо задать различные звуковые файлы для оповещения. В этом случае необходимо в настройках группы включить опцию **Разрешить дополнительную настройку отдельных паспортов**. Тогда, выбрав на вкладке **Паспорта** нужный паспорт, можно задать для него свои настройки воспроизведения.

Для каждой ситуации можно задать звуковой файл или группу файлов, которые будут воспроизводиться при ее срабатывании. Точно так же можно составить и сообщение, которое будет выводиться пользователю при срабатывании ситуации. Это же сообщение будет записано в базу данных событий. Для удобства в сообщении можно использовать специальные теги. Например, задав сообщение «Внимание! %п», мы указываем, что пользователю будет выведено сообщение **Внимание!** и далее – шифр параметра. Эти же самые теги можно использовать и при формировании сообщений для sms и электронной почты.

3. Распределенные системы сигнализации.

Не все аварийные ситуации необходимо контролировать на верхнем уровне. Например, аварийные защиты желательно реализовывать в контроллере. Это повышает надежность и скорость реакции системы: выполнения защитных отключений и блокировок. А поскольку контроль срабатывания защит осуществляет контроллер, то нет смысла повторно анализировать ситуацию на верхнем уровне. Достаточно отследить состояние соответствующей защиты, и в случае ее срабатывания генерировать аларм. Для осуществления такого контроля у каждой сигнализируемой ситуации предусмотрена опция **отслеживать ситуацию по паспорту**. Задав паспорт, соответствующий защите, нам остается только ждать, когда защита сработает, и затем проинформировать пользователя в соответствии с настройками (рис. 4).

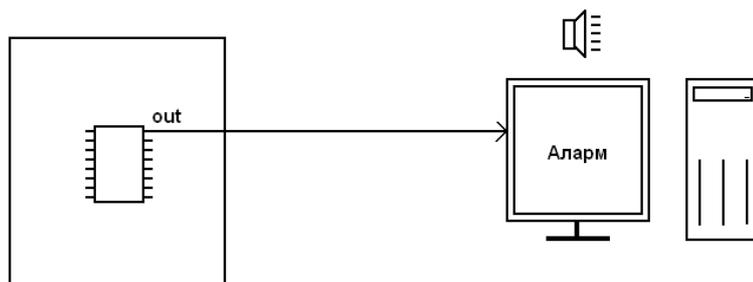


Рис. 4. Отслеживание ситуации по паспорту.

Возможна и обратная ситуация. Возникла аварийная ситуация, модуль АПС сгенерировал аларм, и информацию об этом необходимо передать в контроллер. В этом случае поможет опция **Запись в паспорт при срабатывании**. При срабатывании аларма будет записано заданное значение в заданную переменную контроллера (рис. 5). В нашем примере на щите управления загорится лампочка и зазвучит звуковой сигнал.

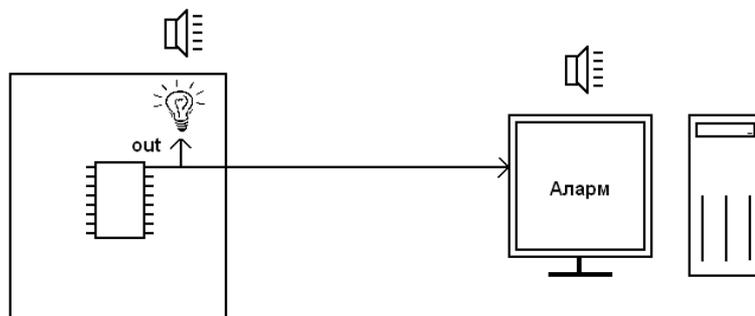


Рис. 5. Запись в паспорт при срабатывании.

Раз уж мы имеем возможность контроллеру о срабатывании аварийной сигнализации и сами получаем аналогичную информацию из контроллера, то, очевидно, необходимо передать в контроллер информацию и о квитировании аларма оператором. В этом поможет опция **запись в паспорт при квитировании** (рис. 6). При квитировании аларма в нужную переменную контроллера будет записано заданное значение. В нашем примере лампочка продолжит гореть, а звуковой сигнал будет снят.

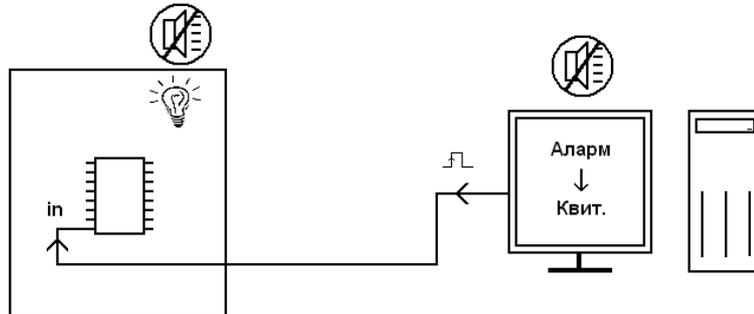


Рис. 6. Запись в паспорт при квитировании.

Поскольку щит может иметь свою местную сигнализацию, то, очевидно, он может иметь и инструмент для квитирования сигнализации там же, по месту. Например, это может быть кнопка **Съём звука**. Оператор нажимает кнопку на щите, инициируя запись в переменную контроллера. Контроллер обрабатывает изменение переменной, квитует аварию и прекращает проигрывание звука. Если отследить с верхнего уровня состояние этой переменной, то можно по команде квитирования по месту сквитировать аларм на верхнем уровне. Для этого нам пригодится опция **квитирование паспортом** (рис. 7).

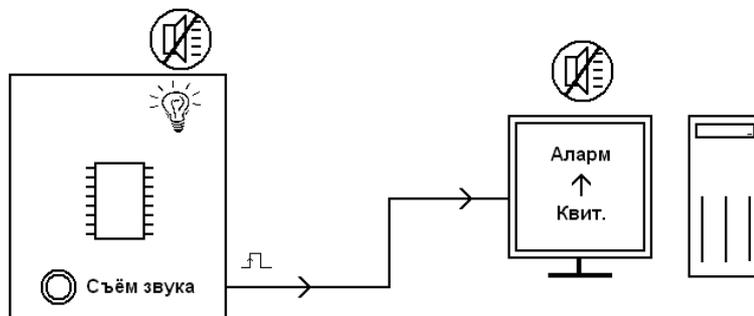


Рис. 7. Квитирование паспортом.

Таким образом, используя перечисленные 4 опции, можно строить распределенные системы сигнализации, которые частично будут обрабатываться на нижнем уровне, а частично на верхнем. Это позволит организовать одновременно и центральную сигнализацию на пульте диспетчера, и местную сигнализации с возможностью квитирования как централизованно, так и по месту.