

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тамбовский государственный технический университет»

**И. А. ЕЛИЗАРОВ, А. А. ТРЕТЬЯКОВ, А. Н. ПЧЕЛИНЦЕВ,
В. А. ПОГОНИН, В. Н. НАЗАРОВ, П. М. ОНЕВСКИЙ**

ИНТЕГРИРОВАННЫЕ СИСТЕМЫ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ: SCADA-СИСТЕМЫ

Допущено Учебно-методическим объединением вузов по образованию
в области автоматизированного машиностроения (УМО АМ)
в качестве учебного пособия для студентов
высших учебных заведений, обучающихся по направлению
подготовки «Автоматизация технологических
процессов и производств»



Тамбов
◆ Издательство ФГБОУ ВПО «ТГТУ» ◆
2015

УДК 681.5:67.02(075)

ББК 3966я73

Е51

Рецензенты:

Доктор технических наук, профессор,
заведующий кафедрой «Компьютерное и
математическое моделирование» ФГБОУ ВПО «ТГТУ»

А. А. Арзамасцев

Доктор технических наук, профессор кафедры
информационных систем ФГБОУ ВПО
«Тверской государственный технический университет»

В. Н. Богатиков

Авторский коллектив:

*И. А. Елизаров, А. А. Третьяков, А. Н. Пчелинцев,
В. А. Погонин, В. Н. Назаров, П. М. Оневский*

Е51 **Интегрированные** системы проектирования и управления:
SCADA-системы : учебное пособие / И. А. Елизаров, А. А. Третьяков,
А. Н. Пчелинцев и др. – Тамбов : Изд-во ФГБОУ ВПО
«ТГТУ», 2015. – 160 с. – 400 экз. – ISBN 978-5-8265-1469-6.

Представлены общие сведения об интегрированных системах управления производством, дан краткий обзор наиболее популярных на отечественном рынке SCADA-систем, описаны все этапы их построения, работа с пакетами «MasterSCADA» и «КРУГ-2000».

Допущено Учебно-методическим объединением вузов по образованию в области автоматизированного машиностроения (УМО АМ) в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки «Автоматизация технологических процессов и производств»

УДК 681.5:67.02(075)

ББК 3966я73

ISBN 978-5-8265-1469-6

© Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет» (ФГБОУ ВПО ТГТУ), 2015

ВВЕДЕНИЕ

Продвижение общества к широкому и всестороннему использованию достижений компьютерной техники и бурно развивающихся информационных технологий требует от студентов – будущих специалистов в области управления технологическими процессами и производствами – знания современных программных средств разработки автоматизированных систем управления технологическими процессами (АСУ ТП).

Системы оперативного диспетчерского управления и сбора данных (системы SCADA – Supervisory Control And Data Acquisition) в настоящее время являются неотъемлемой частью современных АСУ ТП и обеспечивают интерфейс между человеком и системой управления.

Учебное пособие предназначено для студентов, которые хотят получить базовые знания о принципах построения интегрированных систем управления производством, о составе и функциях SCADA-систем. В пособии приведены примеры построения автоматизированных рабочих мест с использованием широко распространённых отечественных SCADA-систем КРУГ-2000 и MasterSCADA.

1. ОБЩИЕ ДАННЫЕ О SCADA-СИСТЕМАХ

1.1. ИНТЕГРИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ПРОИЗВОДСТВОМ

В настоящее время ни одна отрасль промышленного производства не обходится без внедрения автоматизированных систем управления технологическими процессами (АСУ ТП).

Современная АСУ ТП часто входит в состав интегрированной системы управления производством. Интегрированные системы управления производством строятся по иерархическому принципу и включают в себя (рис. 1.1):

- ERP-системы (Enterprise Resource Planning) – планирование ресурсов предприятия (MRP-системы (Manufacturing Resource Planning) – планирование ресурсов производства);
- MES-системы (Manufacturing Execution Systems) – исполнительная система производства;
- SCADA (Supervisory Control And Data Acquisition) – система сбора данных и оперативного диспетчерского управления;
- PLC (Programmable Logic Controllers) – программируемые логические контроллеры (ПЛК).

Программное обеспечение (ПО) уровня 3 представлено SCADA-системами. Термин SCADA обычно относится к централизованным системам контроля и управления всей системой или комплексами систем, осуществляемого с участием человека.

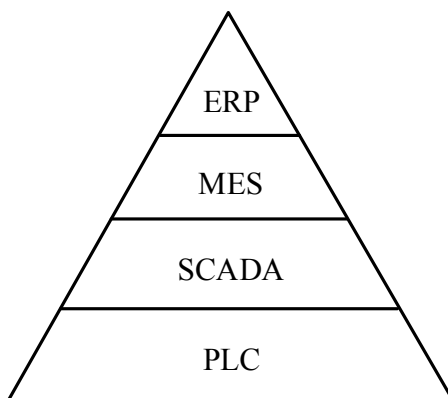


Рис. 1.1. Интегрированные системы управления производством

SCADA-системы реализуют все основные функции визуализации измеряемой и контролируемой информации, передачи данных и команд системе контроля и управления. Данные системы являются основным и в настоящее время наиболее перспективным методом автоматизированного управления технологическими процессами в различных областях. На принципах диспетчерского управления строятся автоматизированные системы управления в промышленности, энергетике, транспорте и отраслях.

SCADA-системы позволяют разрабатывать АСУТП на клиент-серверной или в распределённой архитектуре и реализовывать следующие основные функции [1]:

- сбор текущей информации от ПЛК или других приборов и устройств связи с объектом (УСО), связанных непосредственно или через сеть с пультом оператора (в том числе с использованием протоколов DDE, OPC и др.);
- первичная (вычислительная и/или логическая) обработка измерительной информации;
- архивирование, хранение и дальнейшая обработка информации;
- представление текущей и архивной информации на дисплее (в виде динамизированных мнемосхем, гистограмм, анимационных изображений, таблиц, графиков, трендов, выделение аварийных ситуаций и т.д.);
- отображение и запись аварийных и предаварийных ситуаций;
- печать отчётов и протоколов различной формы;
- ввод и передача команд оператора в контроллеры и другие устройства системы;
- регистрация всех действий оператора;
- защита от несанкционированного доступа и предоставление различных прав пользователям во время работы с системой;
- регистрация всех ошибок и событий внутри системы управления (аппаратные тревоги, ошибки работы сети и т.д.);
- реализация прикладных программ пользователя и их взаимосвязь с текущей измеряемой информацией и управленческими решениями;
- обеспечение информационных связей с серверами и другими рабочими станциями через разные сетевые структуры;
- обеспечение связи с внешними приложениями (СУБД, электронные таблицы, текстовые процессоры и т.д.).

К SCADA-системам предъявляются следующие основные требования [2]:

- надёжность системы (технологическая и функциональная);
- безопасность управления;
- точность обработки и представления данных;
- простота расширения системы.

Использование SCADA-систем позволяет существенно повысить эффективность производственного процесса за счёт [1]:

- более точного ведения технологического процесса, стабилизации качества продукции и уменьшения процента брака;
- уменьшения действий оператора, с целью концентрации его внимания на выработке более эффективных решений по управлению процессом;
- программного контроля правильности выработки команд дистанционного управления и, следовательно, минимизации количества ошибок, допускаемых операторами;
- автоматического выявления и оповещения об аварийных и предаварийных ситуациях;
- предоставления полной необходимой информации персоналу в виде различных отчётов;
- анализа факторов, влияющих на качество готовой продукции.

1.2. СОСТАВ SCADA-СИСТЕМ

Все современные SCADA-системы включают три основных структурных компонента (рис. 1.2) [3, 4]:

- Remote Terminal Unit (RTU) – удалённый терминал, подключающийся непосредственно к контролируемому объекту и управление в режиме реального времени. К RTU можно отнести датчики, исполнительные механизмы, УСО, микроконтроллеры, осуществляющие обработку информации и управление в режиме жёсткого реального времени.
- Master Terminal Unit (MTU), Master Station (MS) – диспетчерский пункт управления (терминал), который осуществляет обработку данных и управление высокого уровня и обеспечивает человеко-машинный интерфейс между человеком-оператором и системой. Реализуется в виде автоматизированного рабочего места (АРМ) оператора (технолога и т.п.).
- Communication System (CS) – коммуникационная система (каналы связи) между RTU и MTU. Она необходима для передачи между удалёнными точками (RTU) и диспетчерским пунктом управления (MTU). В качестве коммуникационной системы можно использовать различные проводные и беспроводные каналы связи.

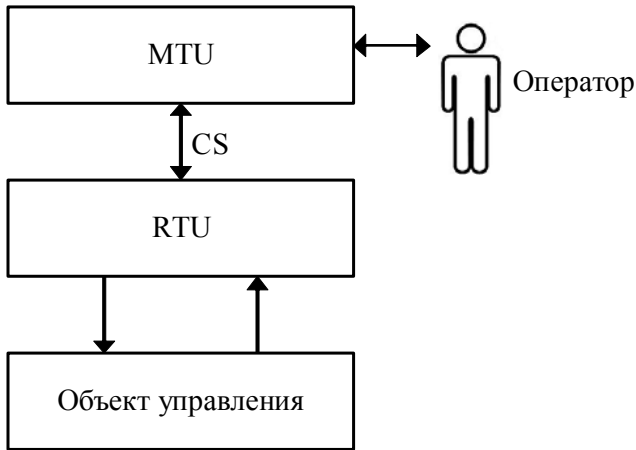


Рис. 1.2. Основные структурные компоненты SCADA-системы

SCADA-система обычно содержит следующие подсистемы:

- драйверы или серверы ввода-вывода (СВВ) данных, обеспечивающие связь SCADA с ПЛК, АЦП и другими устройствами ввода/вывода информации;
- система реального времени, позволяющая выполнять приложения с распределёнными приоритетами;
- НМІ – человеко-машинный интерфейс, позволяющий визуализировать данные о ходе процесса оператору, и позволяет оператору контролировать процесс и дистанционно управлять им;
- база данных реального времени, обеспечивающая хранение информации о процессе в режиме реального времени;
- система логического управления, обеспечивающая исполнение пользовательских программ в SCADA-системе;
- система управления тревогами, обеспечивающая автоматический контроль технологических событий, а также обработку событий оператором или компьютером;
- генератор отчётов, обеспечивающий создание пользовательских отчётов о технологических событиях;
- редактор для разработки человеко-машинного интерфейса;
- редактор для разработки пользовательских программ;
- внешние интерфейсы обмена данными между SCADA и другими приложениями (OPC, DDE, ODBC, DLL и т.д.).

Для связи с аппаратурой нижнего уровня современные SCADA-системы предоставляют большой набор драйверов или серверов ввода/

вывода и средств создания собственных программных модулей или драйверов новых устройств нижнего уровня.

Для подсоединения драйверов ввода/вывода к SCADA-системе в настоящее время используются следующие механизмы:

- стандартные протоколы динамического обмена данными (DDE), открытый механизм взаимодействия с базами данных ODBC;
- собственные протоколы фирм-производителей SCADA-систем, реально обеспечивающие самый скоростной обмен данными;
- OPC (OLE for Process Control) – протокол, который является стандартным и поддерживается большинством SCADA-систем.

Недостатками протоколов DDE являются:

- необходимость написания для каждой SCADA-системы своего драйвера для поставляемого оборудования;
- два пакета не могут иметь доступ к одному драйверу в одно и то же время, так как каждый из них поддерживает обмен именно со своим драйвером.

OPC (OLE for Process Control) лишён недостатков протоколов DDE.

Цель OPC-стандарта заключается в определении механизма доступа к данным с любого устройства из приложений [5]. OPC позволяет производителям оборудования поставлять программные компоненты, которые стандартным способом обеспечат клиентов данными с ПЛК.

Преимущество использования OPC:

- OPC позволяет определять на уровне объектов различные системы управления и контроля, работающие в распределённой гетерогенной (неоднородной) среде;
- OPC устраняет необходимость использования различного нестандартного оборудования и соответствующих коммуникационных программных драйверов.

В настоящий момент многие производители, выпускающие SCADA-системы, выпускают версии своих программных продуктов под различные операционные системы (ОС) (Windows, Unix, QNX, VMS и другие операционные системы реального времени (ОСРВ)). Подавляющее большинство SCADA-систем реализовано на платформах MS Windows. Большим недостатком SCADA-систем на платформах Windows по сравнению со SCADA-системами на платформах (ОСРВ) является отсутствие поддержки жёсткого реального времени [5].

Ряд фирм разработали расширения реального времени для семейства платформ Windows NT, превращающие их в ОС жёсткого реального времени (например, подсистеме реального времени RTX (Real Time Extension) фирмы Ventur Com).

Одной из основных задач, решаемых SCADA-системами, является обеспечение НМИ при представлении информации о процессе. Составной частью графического интерфейса являются формы (экраны) с изображением мнемосхем технологического процесса, анимационных изображений, гистограмм, таблиц, и т.д. Для ограничения доступа к изменяемым параметрам системы предусмотрена система защиты от несанкционированного доступа.

Система управления тревогами SCADA-систем предупреждает оператора о возникновении событий, которые могут привести к серьёзным последствиям, и которые требуют реакции оператора. В журнале тревог различают квитированные (**подтверждённые**) и неквитированные (**неподтверждённые**) тревоги. Тревога становится подтверждённой, когда оператор отреагировал на сообщение о тревоге. До этого времени тревога остаётся в состоянии неподтверждённой.

Большинство SCADA-систем содержат встроенные средства разработки приложений пользователей. При этом выделяются два подхода реализации средства разработки:

- использование языков высокого уровня (Basic-подобные языки);
- использование языков международного стандарта МЭК 6-1131/3 (ST (Structured Text), FBD (Function Block Diagram), SFC (Sequential Function Chart), LD (Ladder Diagram), IL (Instruction List)).

Языки МЭК 6-1131/3 обладают высокой функциональностью и простотой использования, при этом предохраняют разработчика программ от многих ошибок, возникающих при использовании языков программирования высокого уровня. Кроме этого реализации языков МЭК 6-1131/3 в SCADA-системах включают расширенные наборы библиотек, реализующих различные алгоритмы управления.

Кроме перечисленных выше подсистем многие SCADA-системы содержат модуль удалённого контроля и управления за ходом технологического процесса с использованием Web-технологии. В этом случае используется технология так называемого «тонкого клиента». При подключении соответствующего модуля и связи с сервером (SCADA-системой) реализуется загрузка в локальный компьютер (клиент) данных о работающем в системе проекте. Вся обработка данных происходит на удалённом сервере, на компьютере-клиенте локальном идёт визуализация данных, используя различные Web-технологии (Java, ActiveX, CGI и т.д). Примером таких наборов подключаемых модулей являются WebClient (US Data, FactoryLink (MonitorPro)), WebActivator (AdAstra, TraceMode), WebКонтроль (НПФ КРУГ, КРУГ-2000), MSRT-Inet (inSAT, MasterSCADA).

Таким образом, гибкость и масштабируемость систем диспетчерского управления в значительной степени достигается модульным строением SCADA-систем, что позволяет поэтапно увеличивать функциональную и информационную мощность [6].

1.3. ОБЗОР SCADA-СИСТЕМ, ПРЕДСТАВЛЕННЫХ НА РЫНКЕ

В настоящий момент времени на мировом рынке представлено большое количество программных пакетов SCADA. Они различаются как по области применения, так и по функциональному составу и информационной мощности. В таблице 1.1 приведены имеющиеся на мировом рынке и использующиеся при проектировании систем автоматизации SCADA-пакеты.

Рассмотрим более подробно некоторые из SCADA-пакетов, представленных в таблице.

InTouch. Scada-система InTouch – пакет для разработки HMI интерфейса и АСУТП, входящий в программный набор FactorySuite корпорации Wonderware (США). FactorySuite – это ПО для разработки ПК АСУ ТП, которое включает в себя различные пакеты для решения различных задач в области промышленной автоматизации.

SCADA-система Intouch – один из популярных в мире программных пакетов создания человеко-машинного интерфейса (HMI) (рис. 1.3). Первые его разработки Wonderware Intouch появились около 20 лет назад.

1.1. Популярные SCADA-пакеты

Наименование SCADA-пакета	Изготовитель
FactoryLink	United States DATA Co, США
InTouch	Wonderware, США
Genesis	Iconics, США
WinCC	Siemens, Германия
Citect	Shneider Electric, США
Trace Mode	AdAstra, Россия
Master-SCADA	ИнСАТ, Россия
КРУГ-2000	КРУГ, Россия

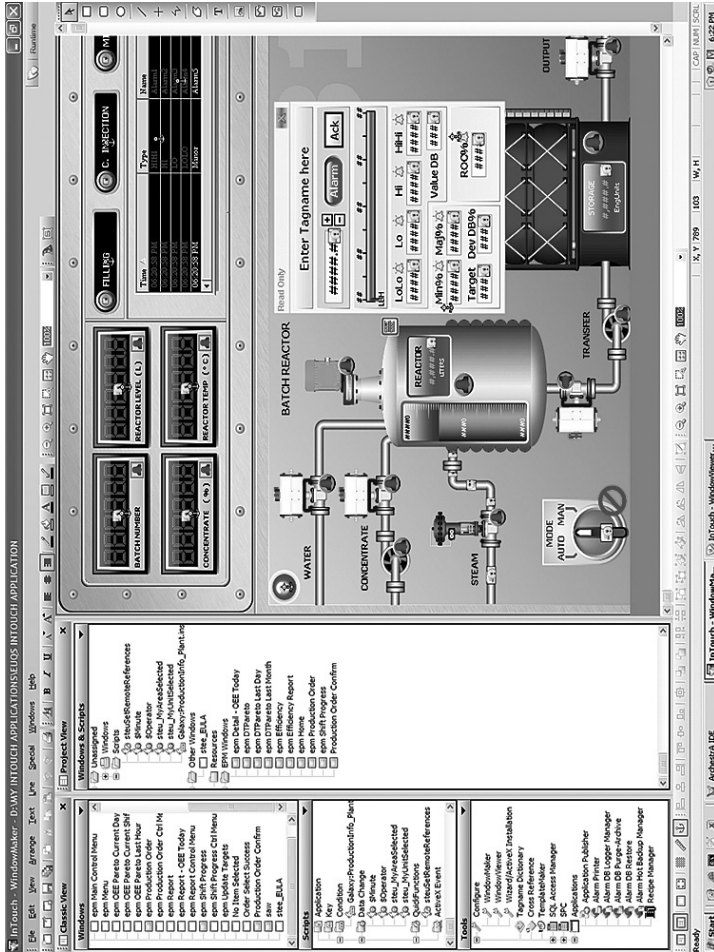


Рис. 1.3. Внешний вид среды разработки приложений WindowMaker Intouch

WindowMaker Intouch. SCADA-система Intouch поддерживает интеграцию со всеми крупными производителями рынка систем и решений автоматике, включая ABB, Siemens и многими другими. Приложения, созданные в этой среде, используются в различных отраслях: от добычи нефти до производства продуктов питания.

SCADA-система Wonderware Intouch состоит из двух частей: среды разработки и среды исполнения. В среде разработки WindowMaker разрабатывают мнемосхемы и описывают сценарии, после чего загружаются в среду исполнения WindowViewer.

Сервер ввода/вывода данных SCADA-системы Intouch ориентирован на конкретное оборудование, для этого используются стандартные протоколы обмена данными: DDE, OPC, TCP/IP и т.д. Информационная мощность системы составляет более чем 120000 сигналов и параметров (тэгов).

В Intouch встроены исторические и временные тренды, элементы управления Microsoft ActiveX и .NET.

Достоинством SCADA-системы Intouch является:

- высокая скорость работы благодаря механизму, который динамически регулирует скорость опроса входных сигналов (опрос происходит только при изменении значения контролируемого параметра);
- архитектура клиент-сервер для эффективной работы в сети с БД;
- открытость – возможность добавлять и использовать компоненты других фирм вследствие поддержки технологий ActiveX и OPC;
- интеграция с другими программными пакетами фирмы Wonderware и простой обмен данными с программными пакетами Microsoft Office (Excel, Access и др.), Microsoft Visual Basic и др.;
- большое число серверов ввода/вывода – более 600;
- возможность создания собственных библиотек алгоритмов;
- автоматический контроль качества сигналов с датчиков и контроллеров;
- распределённая система обнаружения и регистрации аварийных ситуаций одновременно поддерживает множество серверов (провайдеров) аварийных ситуаций, что даёт возможность операторам видеть информацию об авариях во многих удалённых местах синхронно.

Citect. CitectSCADA – программный пакет компании Shneider Electric (США), представляющий собой полнофункциональную систему мониторинга, сбора данных и управления, которая обеспечивает:

- графическую визуализацию процесса;
- управление тревогами (алармами – предупреждающими сообщениями);
- доступ к архивным трендам;

- отслеживание трендов в реальном времени;
- подготовку отчётов;
- многопоточное выполнение подпрограмм пользователя, разработанных на CitectVBA и CiCode.

Особенностью данной SCADA-системы является то, что среда разработки Citect поставляется бесплатно, оплачиваются только исполнительные (runtime) лицензии. Это дает возможность пользователю разработать и протестировать пробный проект, не тратя значительных средств на начальном этапе. При лицензировании SCADA-системы Citect учитывается не общее число компьютеров, на которых установлен Citect, а число одновременно работающих компьютеров.

SCADA-система Citect построена на базе мультизадачного ядра реального времени, обеспечивающего высокую производительность работы с большими объёмами данных, т.е. увеличение количества параметров незначительно изменяет время отклика системы.

SCADA-система Citect применяется для автоматизации как небольших систем с десятками (сотнями) параметров, так и для больших систем с сотнями тысяч параметров. Эта масштабируемость определяется модульной клиент-серверной архитектурой, в которой каждый функциональный модуль SCADA-системы Citect может исполняться на отдельном компьютере и даже быть распределён на несколько компьютеров для увеличения общей производительности.

SCADA-система Citect состоит из пяти функциональных модулей (серверов или клиентов):

- I/O – сервер ввода/вывода, обеспечивающий передачу данных между физическими устройствами ввода/вывода и остальными модулями Citect;
- Display – клиент визуализации, обеспечивающий операторский интерфейс: отображает данные, поступающие от других модулей Citect и управляет выполнением команд оператора;
- Alarms – сервер алармов (тревог), отслеживающий данные, сравнивает их с допустимыми пределами, проверяет выполнение заданных условий и отображает алармы на соответствующем узле визуализации;
- Trends – сервер трендов. Собирает и регистрирует трендовую информацию, позволяя отображать развитие процесса в реальном масштабе времени или в ретроспективе;
- Reports – сервер отчётов. Генерирует отчёты по истечению определённого времени, при возникновении определённого события или по запросу оператора.

Каждый функциональный модуль Citect исполняется как отдельная задача независимо от того, исполняются ли модули на одном компьютере или на разных. Поэтому SCADA-система Citect позволяет строить как простые системы, когда все модули работают на одном компьютере, так и сложные, в которых функциональные модули распределены по отдельным узлам локальной сети частично.

Для разработки графического интерфейса SCADA-система Citect используются библиотеки простых графических объектов (фигуры, линии, трубы, точечные изображения, текст и др.) и технологических символов (насосы, механизмы, резервуары и др.) (рис. 1.4). Любой объект или символ могут быть анимированы любым доступным в SCADA-системе Citect способом.

Предупреждающие сообщения (тревоги) передаются оператору в специализированных окнах, называемых страницами или через анимлируемые графические объекты, например, смена цвета объекта при возникновении аварийной ситуации. С каждой тревогой можно связать определённое действие, которое будет выполняться при появлении данного события, например, воспроизвести звуковой файл. Информация о тревогах и реакция на них оператора может выводиться как автоматически при возникновении тревоги, так и по запросу оператора.

В Citect может выводиться на экран и регистрироваться любой производственный параметр. В трендах системы может отображаться несколько переменных, которые на экране дают графическое представление о ходе процесса во времени. Для создания трендов в системе можно использовать готовые шаблоны. Необходимые выборки извлекаются периодически или (и) в момент возникновения в системе определённых событий.

Отчёты в Citect формируются в виде документов, отражающих заданные производственные показатели и выдаваемых по запросу, периодически, либо при возникновении определённого события. Отчёты генерируются в любом удобном для пользователя формате и могут выводиться на экран, распечатываться, а также сохраняться на диске для последующего использования.

SCADA-система Citect включает развитый язык программирования CiCode, который содержит более 1000 функций и позволяет создать прикладную программу пользователя любой сложности. Язык CiCode поддерживает большое количество операторов для управления алармами, операторов для работы с файлами, операторов для организации работы с трендами, операторов для управления коммуникационными портами, SQL-функции и множество других функций и операторов.

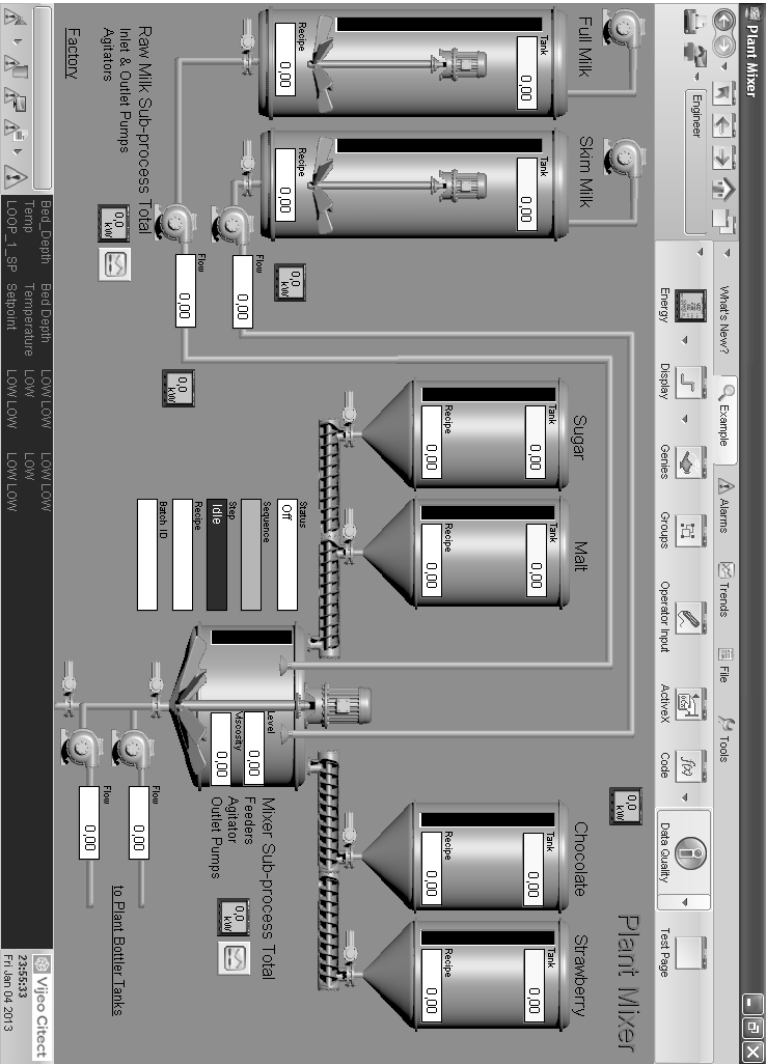


Рис. 1.4. Мнемосхема технологического процесса в SCADA-системе Citect

Trace Mode. SCADA-система Trace Mode компании AdAstra Research Group, Ltd (Россия) – высокотехнологичная программная система для автоматизации технологических процессов, телемеханики, диспетчеризации, учёта ресурсов и автоматизации зданий. Это интегрированная информационная система для управления промышленным производством, объединяющая в едином целом продукты класса SOFTLOGIC-SCADA и экономические модули T-Factory HMI-MES-EAM.

Интегрированная платформа для управления производством Trace Mode состоит из Интегрированной среды разработки (объединяет более 10 различных редакторов проекта), в которой осуществляется создание проектов АСУ и из набора исполнительных модулей, обеспечивающих функционирование системы в реальном времени [7]. Интегрированная среда разработки включает полный набор средств построения АСУ ТП, а именно средства создания:

- операторского интерфейса (SCADA/HMI);
- распределённых операторских комплексов;
- промышленной базы данных реального времени;
- программ для промышленных контроллеров (SOFTLOGIC).

Достоинством программного пакета Trace Mode является большая библиотека встроенных драйверов, которая поставляется бесплатно. Система поддерживает большое число как отечественного, так и зарубежного оборудования и позволяет разрабатывать высоконадёжные распределённые АСУ ТП.

Удобным инструментом создания проекта в SCADA-системе Trace Mode является технология автопостроения [8]. Она позволяет быстро создать связи между узлами распределённой системы управления, между источниками данных SCADA и каналами, создать источники данных по известной конфигурации контроллера и т.п.

Для создания прикладных программ пользователя в SCADA-системе Trace Mode поддерживаются 5 языков международного стандарта IEC 61131-3. Это процедурные языки – Techno ST, Techno IL и визуальные языки – Techno FBD, Techno LD, Techno SFC. Все языки программирования снабжены мощными средствами отладки.

Графический редактор SCADA-системы поддерживает трёхмерную графику и достаточно прост в освоении. Любой графический элемент на мнемосхеме может обладать динамикой, т.е. может менять свои свойства, положение на экране и размеры в реальном времени в зависимости от параметров, а также служить кнопкой.

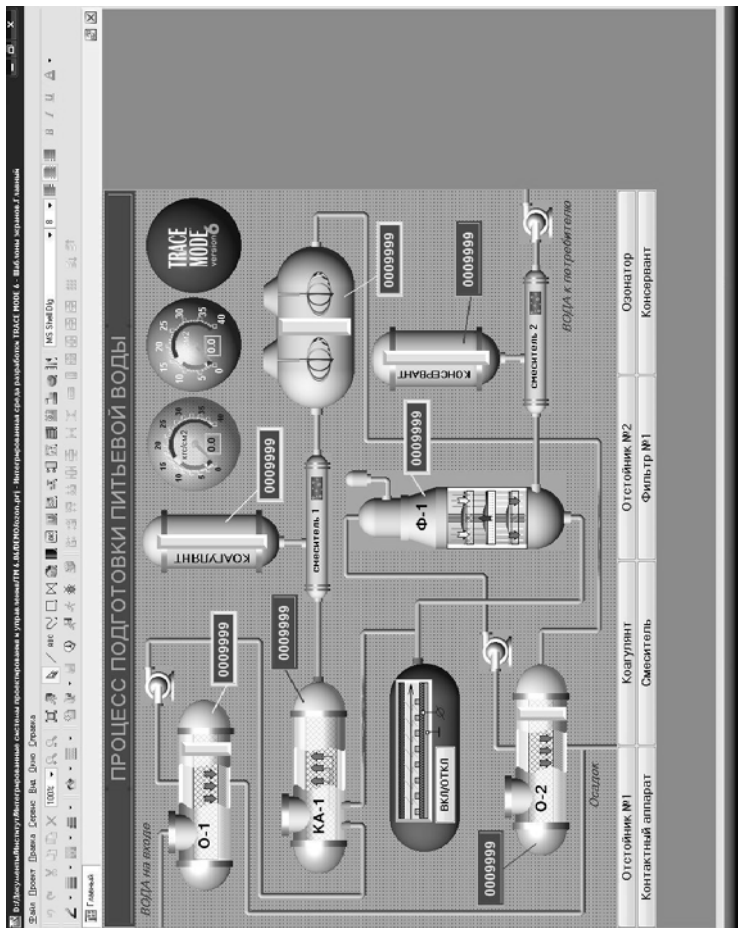


Рис. 1.5. Мнемосхема технологического процесса в SCADA-системе Trace Mode

Объёмные фигуры в Trace Mode обладают рядом уникальных свойств:

- произвольный базовый цвет;
- полупрозрачность;
- наложение текстур;
- коэффициенты отражения и рассеивания света;
- настраиваемое расположение источника света;
- поворот на произвольный угол (в том числе в реальном времени);
- отображение внутренних поверхностей для рисования разрезов ёмкостей.
- фигурные срезы цилиндров и труб.

Для создания сложных мнемосхем в графическом редакторе SCADA-системы поддерживаются слои, видимостью которых можно управлять в реальном времени [SCADA-система TRACE MODE. [9]. Также в системе реализована поддержка всплывающих (Pop-Up) окон. Всплывающие окна располагаются поверх окна основной мнемосхемы и служат вспомогательными элементами.

Для отображения текущей и исторической информации в SCADA-системе Trace Mode реализованы универсальные тренды, которые совмещают функции архивного тренда и тренда реального времени, дискретного и аналогового трендов. Глубина тренда ограничена объёмом накопленного архива встроенной промышленной СУБД SIAD/SQL.

Генератор отчётов SCADA-системы позволяет в реальном времени создавать ясные и полнофункциональные HTML-отчёты. Шаблоны отчётов создаются в Редакторе шаблонов Интегрированной среды разработки.

В СУБД реального времени SIAD/SQL SCADA-системы Trace Mode архивные данные не только хранятся, но и подвергаются статистической обработке в реальном времени, и могут отображаться на мнемосхемах и использоваться в программах наравне с данными реального времени.

SCADA-система Trace Mode обладает широкими возможностями интеграции с базами данных и другими приложениями. Система поддерживает популярные программные интерфейсы: OPC, DDE, ODBC. Взаимодействие с внешними базами осуществляется с помощью встроенного редактора SQL-запросов. Кроме того, существует возможность подключения компонентов ActiveX.

1.4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чём состоит иерархический принцип построения интегрированных систем управления производством?
2. Какие основные функции реализованы в SCADA-системах?
3. Какие основные требования предъявляются SCADA-системам?
4. Какие основные структурные компоненты SCADA-системы?
5. Назовите основные подсистемы SCADA-систем?
6. Какие механизмы используются для подключения драйверов ввода/вывода к SCADA-системе?
7. Какие преимущества даёт использование OPC для реализации механизма доступа к данным?
8. Для каких операционных систем реализованы SCADA-системы?
9. Каким образом реализуется человеко-машинный интерфейс в SCADA-системах?
10. Какие средства разработки приложений пользователей реализованы в SCADA-системах?

2. MASTERSCADА

2.1. ОБЩИЕ СВЕДЕНИЯ

MasterSCADA является одной из лидирующих на российском рынке SCADA-систем. Разработчиком MasterSCADA является ООО «ИнСат» (www.insat.ru, www.masterscada.ru), Москва. Список внедрений MasterSCADA насчитывает уже много тысяч систем, которые успешно работают на территории СНГ.

SCADA-система MasterSCADA – это инструмент разработки АСУ ТП, в котором реализована совокупность средств и методов, обеспечивающих резкое сокращение трудозатрат и повышение надёжности создаваемой системы

Основные преимущества пакета [10]:

1. Единая среда разработки SCADA-системы, которая позволяет:
 - решить проблемы программной стыковки различных устройств системы управления;
 - с лёгкостью перераспределять сигналы или алгоритмы их обработки по отдельным устройствам;
 - создавать распределённые по устройствам алгоритмы контроля и управления;
 - иметь доступ с любого рабочего места к любой информации, имеющейся в системе.
2. Раздельное конфигурирование структуры системы и логической структуры объекта даёт возможность:
 - разрабатывать эти структуры параллельно;
 - независимо работать специалистам различных профилей;
 - решить проблему перехода от одной технической структуры системы к другой.
3. Открытость и следование стандартам, которая обеспечивает:
 - взаимодействие с другими программами с помощью современных технологий (OPC, OLE, DCOM, ActiveX, OLE DB, ODBC и др.);
 - использование в операторском интерфейсе системы документов любого типа и обмен данными с ними;
 - неограниченное расширение функциональности MasterSCADA продуктами сторонних разработчиков;
 - связь с АСУ-производством;
 - открытые интерфейсы для создания пользователем любых базовых элементов.

4. Интуитивная лёгкость освоения. Удобство инструментария:
- простой и понятный русскоязычный интерфейс;
 - реализация большинства действий пользователя методом «перетащи и брось»;
 - подробный справочный материал;
 - запоминание всех индивидуальных настроек;
 - контекстная справка и всплывающие подсказки;
 - контроль допустимости вводимой информации;
 - нераздельное единство SCADA- и SoftLogic-системы;
 - соответствие проекта логике восприятия системы и объекта разработчиком;
 - возможность разработки проекта в удобном порядке;
 - возможность полной отладки проекта без связи с объектом;
 - возможность полной отладки распределённой системы на одном компьютере;
 - отсутствие необходимости настройки сети или выделения отдельного сервера для запуска распределённой системы;
 - возможность многократного использования любой ранее созданной части системы;
 - возможность использования пакета в качестве инструмента моделирования, создания тренажеров и демоверсий.
5. Мощная трёхмерная графика и мультимедиа:
- библиотека объёмных элементов со встроенным индикатором уровня заполнения;
 - стереометрически правильные врезки одних элементов в другие на основании соотношения их диаметров и углов наклона;
 - объёмный трубопровод любой сложности;
 - импорт изображений в любых графических форматах;
 - встроенный инструментарий создания мультипликации;
 - динамизация любых свойств любых ActiveX-объектов без программирования;
 - имитационный режим для проверки настроек анимации.
6. Неограниченная гибкость вычислительных возможностей:
- визуальное создание схемы вычислений на языке функциональных блоков (FBD);
 - программирование алгоритмов на технологическом процедурном языке ST;
 - библиотека из более 150 функциональных блоков, включая контроль и управление;

- первичная обработка каждого сигнала с автоматическим контролем до 6 границ;
- формульные вычисления значений и событий с обширной библиотекой функций;
- автоматическая и пользовательская обработка признаков качества значений;
- автоматическая индикация значений всех вычисленных сигналов;
- имитационный режим с индивидуальным выбором функций имитации сигналов;
- возможность создания пользовательских функциональных блоков и макроблоков;
- интеграция вычислительных, событийных и визуальных функций объектов;
- открытая объектная модель и возможность подключения скриптов, написанных на «С#»;
- открытые интерфейсы для подключения собственных алгоритмов, написанных на языке «С».

7. Объектный подход

Объект в MasterSCADA – это основная единица разрабатываемой системы, соответствующая реальному технологическому объекту (цеху, участку, аппарату, насосу, задвижке, датчику и т.п.), управляемому разрабатываемой с помощью MasterSCADA системой. С другой стороны, это и традиционный с точки зрения программирования объект, обладающий следующими стандартными для программных объектов качествами.

- Объект имеет набор свойств и документов, которые жёстко связаны с ним. Свойства объекта – это, например, период опроса и способ обработки сигналов от его датчиков. Документы объекта – его изображение, описание, чертёж, перечень сообщений и т.п. В MasterSCADA нет просто тренда, рапорта или мнемосхемы: каждый документ в разрабатываемой системе всегда относится к какому-либо объекту, являясь его свойством.

- Ограничение области видимости. Вы можете задать ограничение области видимости. В этом случае для объекта не допускается использование в документах «чужих» данных. «Своими» считаются только собственные входы и выходы или входы/выходы подчинённых объектов. Благодаря этому при сохранении объекта в библиотеке или переносе его в другой проект ничего, кроме внешних связей настраивать не требуется. Все настройки и документы сохраняются.

- Наследование. По умолчанию все настройки наследуются от «родительского» объекта. Каждый объект имеет множество настроек. Такое обилие могло бы потребовать от разработчика системы выпол-

нения огромного количества действий. Но так как для разных объектов их список в основном одинаков, то все настройки можно сделать только один раз, все подчинённые объекты автоматически воспримут настройки родительского элемента, т.е. «унаследуют» их. Исключение будут составлять только те настройки и только у тех элементов, которые разработчик изменил сам.

- Типизация и тиражирование. Допускается многократное использование одного и того же объекта со всеми созданными для него документами, в том числе при разработке различных систем. При копировании объекта или сохранении его в библиотеке все его настройки и документы, настройки документов и внутренние связи будут сохранены. Внешние связи с источниками данных будут восстановлены при наличии источников с такими именами, внешние связи с приёмниками данных будут восстановлены, если эти приёмники данных свободны, остальные будут показаны в общем списке. Благодаря этому управление и контроль типовым технологическим объектом (насосом, задвижкой, реактором, фильтром и т.п.) создаются один раз для всех проектов. Это позволяет создавать объекты для одной системы параллельно независимыми разработчиками.

Архитектура системы автоматизации. MasterSCADA позволяет создавать системы с самой различной архитектурной конфигурацией: клиентсерверной, одноранговой, многоуровневой и др. На любой стадии проекта возможно изменение архитектуры с минимальными изменениями в проекте (автоматизированной перепривязкой параметров) [11].

С точки зрения назначения компьютерных узлов распределённой системы можно выделить серверы ввода/вывода, операторские станции, клиентские станции, архивные серверы баз данных, Интернет-клиенты. Серверы ввода/вывода и операторские станции обмениваются данными с нижним уровнем системы (контроллерами, модулями удалённого ввода/вывода, счётчиками коммерческого учёта, интеллектуальными датчиками и т.п.). Клиентские станции получают информацию от серверов и операторских станций. Для поддержки Интернет-клиентов на них дополнительно должен быть установлен Интернет-сервер. Сами Интернет-клиенты ни в чём, кроме браузера, не нуждаются. Для связи с базами данных в MasterSCADA предусмотрены так называемые БД-коннекторы – модули связи с SQL-совместимыми СУБД.

MasterSCADA – вертикальноинтегрированная система, она позволяет не только реализовать АРМ операторов-технологов, но и запрограммировать управляющие контроллеры, причём распределение и перераспределение программ между компьютерами и контроллерами также возможно на любой стадии разработки проекта. Это связано с тем, что объект в объектной иерархии MasterSCADA принадлежит

конкретному компьютеру и, возможно, связанному с ним контроллеру. Поэтому, если у объекта (например, технологической установки или аппарата) имеется какая-либо логика контроля и управления, то она при наличии связанного с объектом контроллера будет автоматически загружена в него сразу после старта системы в режиме исполнения.

На сегодня поддерживаются контроллеры с операционными системами DOS, MiniOS7, Embedded Windows, Windows CE, Linux, Ecos и процессорами x86, ARM, StrongARM, XScale. Связь с контроллерами возможна по различным каналам: Ethernet, RS232/RS485, GSM (Data, SMS, GPRS), радио, телефон. К последовательным портам контроллеров можно подключать модули удалённого ввода/вывода по протоколам Modbus или DCON (совместимый с сериями ADAM-4000, i-7000 и др.).

MasterSCADA работает не только с контроллерами, запрограммированными в её среде. Для встраиваемых в компьютер плат имеется драйверный интерфейс, а для подключения внешних устройств в пакете реализована технология «OPC в ядре системы», благодаря которой обеспечивается эффективное с точки зрения удобства и быстродействия взаимодействие с внешними OPC-серверами в стандартах OPC DA (данные) и HDA (архивы). В последнем случае полученные по этому протоколу архивы будут автоматически включены в архивы MasterSCADA, а при необходимости переданы во внешний SQL-сервер. При создании гетерогенных систем MasterSCADA может сама выступать в роли OPC-сервера (DA и HDA).

В MasterSCADA предусмотрены специальные средства повышения надёжности и живучести создаваемых систем. Прежде всего, это средства резервирования – разработчик проекта может резервировать компьютеры, контроллеры, сети. Переключение на резерв производится по команде средств диагностики, вычисляемому в пользовательской программе событию или по команде оператора. Для обеспечения безударного восстановления после сбоя и в компьютерах, и в контроллерах предусмотрен горячий рестарт с восстановлением значений параметров и внутренних накопительных ячеек расчётных алгоритмов. В случае перерывов в работе каналов связи с контроллерами они могут накапливать собственные архивы, которые будут целиком переданы в архивы верхнего уровня после восстановления связи.

Подсистема контроля действий операторов предусматривает ведение журналов, а также предоставление доступа к просмотру информации или управлению в соответствии с должностью зарегистрированного в данный момент на компьютере оператора. Настройка ограничений может быть сделана как для всех действий одного типа (например, управления исполнительными механизмами), так и индивидуально (для конкретной команды).

2.2. СРЕДА РАЗРАБОТКИ MASTERSCADA

2.2.1. ИНТЕРФЕЙС СРЕДЫ РАЗРАБОТКИ

Среда разработки MasterSCADA состоит из четырёх частей (рис. 2.1) [12]:

1. Дерево системы.
2. Дерево объектов.
3. Панель свойств элемента проекта.
4. Палитра функциональных блоков.

Пользовательский интерфейс MasterSCADA построен на идеологии «все в одном». Все модули расширения встроены в общую оболочку. Пользователь всегда работает с единым внешним видом программы, состоящим из древовидного проекта, палитры библиотечных элементов и окна редактирования документов и свойств.

Проект состоит из двух частей – **дерево системы** и **дерево объектов**.

Дерево системы описывает структуру создаваемой системы (один или несколько компьютеров), а также содержит источники данных – OPC-серверы, контроллеры, связи с базами данных.

Дерево объектов описывает иерархическую структуру проекта – в данном дереве создаётся программа, реализующая обработку получаемых из дерева системы данных, их преобразование, архивирование и представление оператору в нужном виде.

Рекомендуется строить иерархический проект: от крупных элементов – к мелким (например Завод – Цех – Печь – Клапан – Состояние). Это позволяет использовать **объектный подход** к построению проекта, дающий разработчику такие инструменты как **тиражирование, наследование, шаблоны**. При этом каждый объект может иметь свои настройки и документы (мнемосхемы, окна, тренды, отчёты), собственные параметры архивирования и набор сообщений. Использование объектного подхода к разработке позволяет быстро создавать крупные проекты, с удобной для оператора системой навигации, а также быстро корректировать их.

В поставку MasterSCADA входит набор библиотек, содержащий более ста различных **функциональных блоков**, расположенных в **Палитре ФБ**. Функциональные блоки позволяют решать необходимые разработчику задачи – проводить математическую и логическую обработку, работать со временем и архивом, управлять исполнительными механизмами, осуществлять операции с документами.

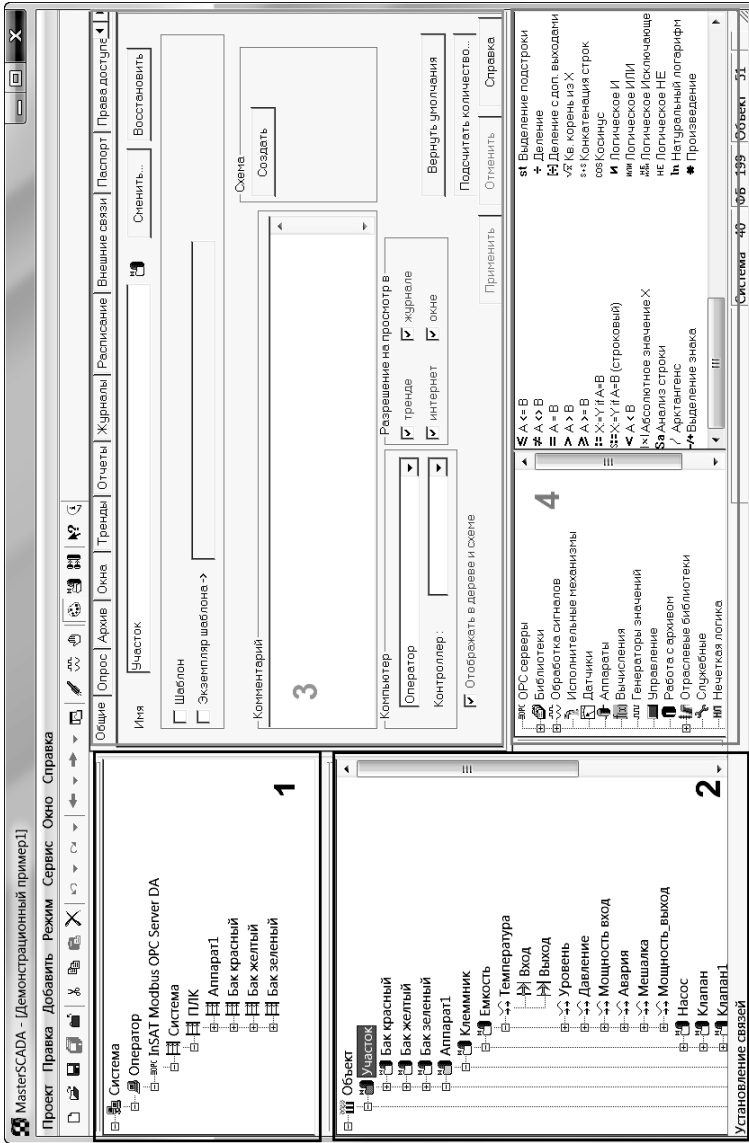


Рис. 2.1. Интерфейс проекта MasterSCADA

Наряду с чисто алгоритмическими функциональными блоками, в MasterSCADA используются *визуальные функциональные блоки (ВФБ)*, которые наряду с алгоритмом имеют отображение на мнемосхемах и окна управления. Каждый ВФБ полностью реализует всё необходимое для контроля и управления тем типом исполнительного механизма или объекта, для которого он был создан.

Существуют библиотеки с общераспространёнными элементами (насосы, клапаны, воздухоудвки), а также библиотеки для определённых отраслей – HVAC (вентиляция и кондиционирование), теплоэнергетика и электроэнергетика.

В MasterSCADA предусмотрена возможность реализации алгоритмов с помощью встроенных редакторов различных языков: **FBD**, **ST** (стандарт МЭК 61131-3) и **C#**. Язык **C#** за счёт доступа к внутренним функциям (API) MasterSCADA можно также использовать для автоматизации проектирования или создания сценариев работы в режиме исполнения. Кроме того, пользователь может пополнить набор функциональных блоков собственными разработками на профессиональных языках программирования (C, C++, C#).

Дерево Системы. Изучая дерево Системы готового проекта можно определить, какую архитектуру реализовал разработчик, какое количество компьютеров будет взаимодействовать в системе, какой способ был выбран для подключения внешних устройств, модулей ввода/вывода, является ли система вертикально-интегрированной (т.е. контроллеры тоже программируются в этом же проекте), какое количество внешних сигналов MasterSCADA будет обрабатывать. По сути, дерево Системы – это своеобразный вид структурной схемы автоматизации [13].

Построение дерева Системы начинается с элемента «Компьютер». Физический компьютер даже в самом простом проекте выполняет несколько функций, например:

- к нему подключается оборудование нижнего уровня;
- он является рабочим местом оператора, технолога и т.п.;
- он передает информацию в приложения других компьютеров.

Таким образом, элемент «Компьютер» – это точка пересечения реальных элементов создаваемой системы автоматизации и виртуальных. Без этого элемента не может существовать ни один проект MasterSCADA. Компьютер может содержать дочерние элементы, с которыми взаимодействует MasterSCADA, установленная на данном реальном компьютере: «Контроллер», «Модуль ввода/вывода», «ОПС-сервер», «MasterLink», «БД-коннектор».

При выборе пункта «Вставить/Контроллер» контекстного меню «Компьютера» имеется возможность вставить в дерево системы контроллер, программируемый средствами MasterSCADA – MasterPLC-контроллер.

Если контроллер запрограммирован не средствами MasterSCADA (а, например, с помощью фирменного ПО производителя), то чтобы обеспечить его работу с MasterSCADA, потребуется OPC-сервер. Элемент «OPC DA сервер» используется для работы с внешними физическими устройствами, модулями ввода/вывода, контроллерами, а также с различными программами по стандарту OPC DA.

Если из прибора нужно получить архивные данные, то необходимо остановить свой выбор на OPC-сервере, который поддерживает стандарт OPC HDA. При выборе архивного OPC-сервера в дерево системы добавляется элемент «OPC HDA сервер». Как правило, этот элемент используется для решения задачи получения достоверных (в том числе с точки зрения точного указания метки времени события, или измерения значений) данных в системах коммерческого и технического учётов.

Элемент «БД-коннектор» служит для организации работы с такими базами данных как MSSQL, Oracle, Sybase, Interbase/Firebird, MySQL, DB2 и др. В настоящий момент, используя элемент «БД-коннектор» можно реализовать следующие задачи:

- хранение архивов MasterSCADA во внешней базе данных;
- экспорт данных;
- чтение и запись в базу данных значений переменных, используя хранимые процедуры;
- получение данных для формирования отчётов с помощью SQL-запросов.

Дерево объектов. Дерево Объектов отражает реальный технологический объект.

Объект в MasterSCADA – это основная единица разрабатываемой системы, соответствующая реальному технологическому объекту (цеху, участку, аппарату, насосу, задвижке, датчику и т.п.), управляемому разрабатываемой с помощью MasterSCADA системой. С другой стороны, это и традиционный с точки зрения программирования объект, обладающий стандартными для программных объектов качествами.

Объект может внутри себя содержать другие объекты, а также переменные и функциональные блоки (служебные библиотечные объекты, предназначенные для контроля и управления, – иногда они соответствуют объектам реального мира, например, насос или задвижка,

а иногда выполняют только одну функцию контроля или управления, например, регулятор) (рис. 2.1). Любой из объектов MasterSCADA имеет свойства и документы, представляющие его для оператора. Можно создавать свои окна с динамической графикой (мнемосхемы), графики изменения параметров во времени (тренды), отчёты, журналы сообщений и другие документы.

Переменные дерева объектов разделяют на 4 типа: значение, расчёт, событие и команда.

Расчёт – используется для обработки данных с помощью математических формул, включающих, кроме действий и функции из обширной встроенной библиотеки. Наряду с выполнением математических и логических (включая побитовые) действий над переменными (в том числе и с их архивными значениями), расчёт позволяет работать со временем, с признаками качества и нарушения границ переменных. Формула обработки в текстовом виде и список обрабатываемых переменных визуально задаются на странице свойств «Формула», а у переменной «Расчёт» в дереве объектов в режиме исполнения показывается результат.

Значение – входная переменная MasterSCADA. С ней можно связать для передачи в неё данных любую выходную переменную. Применяется только для единичного использования переменной в проекте – для расчётов и отображения полученной информации на трендах и мнемосхемах. Если же перед отображением на мнемосхеме или тренде необходимо произвести обработку данных, или использовать полученное значение в нескольких местах проекта, то данную переменную лучше заменить функциональным блоком «Повторитель», который находится в категории палитры MasterSCADA «Обработка сигналов».

Команда – переменная, предназначенная для передачи значения, заданного оператором или присвоенного ей в результате выполнения действий в другие переменные – например, в значения, выходы (в дереве системы) или расчёты. С помощью команды можно вводить уставки, включать или отключать оборудование, менять коэффициенты в формулах расчётов и т.п.

Событие – отличается от расчёта только тем, что результатом вычислений будет не число, а факт изменения логического значения из состояния Ложь (значение 0) в состояние Истина (значение 1). Момент перехода порождает необходимые действия или выдачу сообщения, заданных разработчиком на страницах свойств этой переменной. Событие имеет формулу, как расчёт, а также страницу свойств «Действия», как команда, но сообщение есть только у события. Имен-

но поэтому основное назначение события – это порождение действий и сообщения при обнаружении какого-либо реального события в реальном мире.

Функциональные блоки. Наряду с переменными MasterSCADA для создания логики работы проекта, обработки сигналов и решения других задач используют функциональные блоки (ФБ).

Функциональный блок имеет набор входов, выходов и встроенную логику работы, т.е. в режиме исполнения элемент обрабатывает поступившие на его входы данные по заданной программе, а результат отображается на его выходах. Некоторые функциональные блоки имеют также своё динамическое отображение в графических окнах (мнемосхемах), список генерируемых сообщений и ряд других свойств объектов. Такие функциональные блоки называются визуальными функциональными блоками (ВФБ).

В палитре функциональных блоков они разделены на несколько групп:

- «Обработка сигналов» и «Вычисления». ФБ этих групп служат для обработки аналоговых, дискретных и строковых величин.

- «Исполнительные механизмы». В этой группе нашли своё место наиболее часто встречаемые в АСУТП исполнительные механизмы: задвижки, клапаны, насосы и т.п. Они представляются как визуальные функциональные блоки.

- «Датчики» и «Аппараты» содержат ВФБ, которые удобно использовать при создании мнемосхем.

- «Работа со временем». ФБ этой группы помогут определить метку времени переменной, выдать импульс в определённый момент времени и много другое.

- «Управление». ФБ этой группы используются в системах, где MasterSCADA выполняет функции регулирования и управления.

- «Работа с архивом» содержит ФБ, которые предоставляют возможности записи и чтения из файлов, обращения к любому значению, сохранённому MasterSCADA.

- «Служебные». Эта группа содержит ФБ, которые позволяют сохранить какие-либо документы проекта или запустить внешнее приложение из интерфейса MasterSCADA, расширить стандартные возможности MasterSCADA при помощи языков программирования ST (стандарт МЭК 61131-3) или C# (ФБ Скрипт).

- «Отраслевые библиотеки» включает в себя ФБ, которые находят широкое применение в отраслях «Теплоэнергетика», «Вентиляция и кондиционирование», «Электроэнергетика».



Рис. 2.2. Палитра функциональных блоков

Базовые документы объектов. В пакете MasterSCADA для каждого объекта можно создавать мнемосхемы (графические окна), тренды, журналы сообщений, отчёты (рапорты). MasterSCADA имеет, как предопределённые типы документов, так и возможность создавать на базе некоторых из них (мнемосхем, журналов сообщений) типизированные новые виды окон пользовательского интерфейса. На странице свойств «Окна» корневого узла дерева системы определяется полный список типов графических документов и их базовые настройки: размер по умолчанию (задаётся в пикселях), способы открытия и отображения окна в режиме исполнения.

Когда мы захотим сделать графическое окно для конкретного объекта, то на его странице свойств «Окна» мы сможем выбрать нужный (по свойствам – размеру, способу вызова и отображения) тип окна. Изначально задано несколько типов: Мнемосхема, Окно объекта, Окно управления и др. документы. Всегда имеется возможность расширить изначальный список, вернувшись на страницу «Окна» дерева системы.

Наиболее часто разработчик начинает с создания мнемосхемы. Для того чтобы создать мнемосхему объекта, необходимо перейти на страницу свойств «Окна», выделить левой клавишей мыши в таблице нужный документ и нажать кнопку «Создать» (рис. 2.3). Если мнемосхема уже была создана ранее, то, чтобы её открыть, нажмите кнопку «Редактировать».

2.2.2. РЕДАКТОР МНЕМОСХЕМ MASTERSCADA

Основным средством взаимодействия с оператором являются мнемосхемы – окна с представлением информации в графическом виде. На мнемосхемах отображаются состояние исполнительных механизмов и аппаратов, значения параметров системы, аварии и т.д. [14].

Мнемосхемы в MasterSCADA принадлежат объекту. Число мнемосхем в проекте не ограничено. Число элементов на мнемосхеме также не ограничено. Библиотеки стандартных элементов содержат множество элементов, включая объёмные элементы со встроенным индикатором заполнения, элементы для создания пользовательских диалогов, элементы, воспроизводящие полный комплект приборов щитового контроля и управления.

Имеется встроенный редактор для создания мультфильмов (с регулируемой прозрачностью изображения) с различными законами трансформации исходных графических файлов (покадровый показ, прокрутка в любом направлении, изменение резкости или размера и т.п.).

Поддерживаются основные мультимедийные форматы: avi, gif, jpg, png, gif, tiff, bmp.

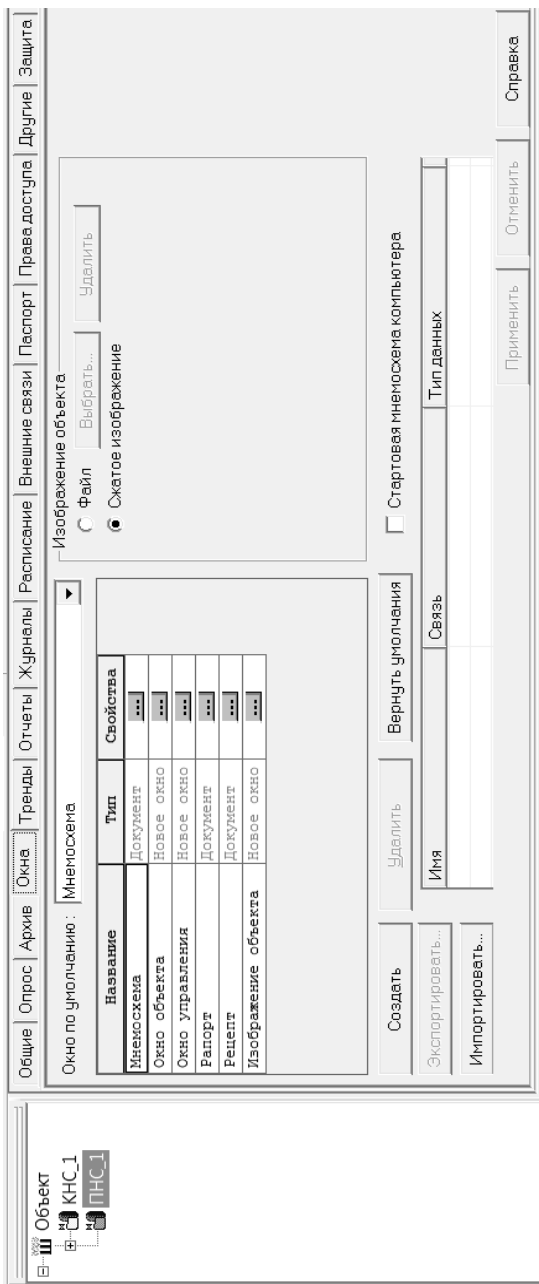


Рис. 2.3. Страница свойств объекта «Окна»

Объёмные трубопроводы произвольной конфигурации создаются в несколько щелчков мыши.

Основной способ создания мнемосхем – перетаскивание элементов из дерева объектов: как включённых в проект из библиотек визуальных функциональных блоков и переменных, уже обладающих всей необходимой функциональностью (динамизированное изображение, окно управления и т.п.), так и созданных пользователем объектов со своими изображениями и окнами управления.

В мнемосхему могут быть перетащены три вида элементов проекта: переменные, визуальные функциональные блоки, объекты. Все эти элементы не требуют для выполнения своих стандартных динамических функций дополнительных настроек.

Например, для переменной может быть выбран способ отображения: в виде текстового значения или в виде одного из типовых приборов щитового монтажа.

Переменные могут быть представлены на мнемосхемах в виде значений или щитовых приборов (способ отображения может быть выбран при перетаскивании переменной с помощью правой кнопки мыши). Значения и приборы уже сопоставлены со шкалой параметра и могут не только показывать числом, столбиком или стрелкой само значение параметра, но и отображать цветом нарушения заданных в шкале границ.

Визуальные функциональные блоки (ВФБ) – это стандартные библиотечные элементы MasterSCADA: исполнительные механизмы, технологические аппараты, элементы управления. Такие ВФБ при перетаскивании из дерева объектов на мнемосхему отображают заданное в них разработчиком графическое представление. Ряд ВФБ позволяет выбрать одно представление из нескольких (ВФБ «Задвижка», «Насос» и ряд других) или выбрать отдельное изображение для каждого состояния (ВФБ «Двухпозиционный механизм с выбором изображения»), либо выбрать отображаемый динамический мультфильм (ВФБ «Динамический ИМ»). Все они имеют окно управления с контролем прав доступа, способ отображения открытого (включённого) и закрытого (отключённого) состояния, индикацию аварии [15].

Объект может быть представлен на мнемосхеме в виде кнопки с текстом, либо уменьшенным изображением его мнемосхемы, также существует возможность создания изображений объекта, который можно иначе назвать «виджетом», т.е. элементом, способным не только открывать мнемосхемы объектом, но и отображать актуальную информацию о состоянии объекта (например – наличие аварий, значение

критических параметров, состояние главных исполнительных механизмов).

Также в качестве элемента мнемосхемы может быть использован любой элемент управления ActiveX, находящийся в графической палитре MasterSCADA, с возможностью динамизации любого его свойства, а также одного из стандартных свойств (положения, размера, отрисовки, мигания и т.п.). Связь с деревом проекта в этом случае осуществляется через переменные динамизации. Этот способ динамизации является менее предпочтительным, чем перетаскивание элементов из дерева объектов.

Если вам не хватает стандартных библиотечных объектов MasterSCADA, то правильнее всего создать новый, сформировав новый объект с необходимой обработкой данных и динамическим изображением, а затем поместив его в библиотеку. Внутренние связи между переменными объекта и динамическими свойствами его графического представления сохраняются, а привязка к проекту сводится к установлению только внешних связей с источниками и приёмниками данных. Это гораздо удобнее, чем типизация на основе создания библиотечных объектов непосредственно в графической палитре MasterSCADA.

Чрезмерная динамизация мнемосхемы может повлечь за собой существенное снижение быстродействия работы MasterSCADA.

Факторы, влияющие на быстродействие:

- Большая площадь перерисовки. Чем больше площадь перерисовки (перерисовывается прямоугольник под любым динамическим элементом), тем больший размер оперативной памяти необходим (если ОЗУ не хватает на хранение элементов, то Windows использует свободную виртуальную память на винчестере – это вызывает постоянную подкачку с диска, так называемый «своп» (swapping), что заметно тормозит систему).

- Используются динамизированные линии и трубопроводы, имеющие более одного вертикального или горизонтального отрезка (линии и трубопроводы, как и другие элементы, занимают площадь описывающего их прямоугольника – поэтому площадь прямоугольника с горизонтальным основанием, описанного вокруг наклонного отрезка, многократно больше собственной площади этого отрезка).

- При сохранении мнемосхемы происходит её оптимизация. Она заключается в том, что все статические элементы переводятся в фоновый рисунок и в режиме исполнения не загружаются. Статическими являются элементы, не реагирующие на действия пользователя и

не меняющие своего изображения при работе. Если статический элемент перекрывает некоторый динамический, то этот статический элемент считается также динамическим и не переводится в фон. В связи с этим настоятельно рекомендуется все динамические элементы размещать на переднем плане, а статические – на заднем.

- Используются ресурсоёмкие динамические элементы изображения (прежде всего, мультфильмы, особенно в формате AVI, во вторую очередь объёмные элементы).

- Мультфильмы имеют большое число кадров и/или большой темп воспроизведения (несколько кадров в секунду), и окно их просмотра имеет большой размер.

- На мнемосхеме используются «тяжёлые элементы» (тренд, журнал, веб-браузер), которые в данный момент обрабатывают данные для отображения.

Динамизацией надо пользоваться очень сдержанно не только из-за потенциального снижения быстродействия работы MasterSCADA, но и из уважения к труду оператора (диспетчера, машиниста). Дело в том, что человек не может постоянно смотреть на двигающееся или мигающее изображение. Неизбежно притупляется внимание. Поэтому настоятельно рекомендуется придерживаться принципа «тёмного щита», когда ярко или, тем более, динамически отображаются только те элементы мнемосхемы, которые требуют внимания оператора «здесь и сейчас».

2.2.3. ТРЕНДЫ MASTERSCADA

Важным элементом SCADA системы является система представления архивных данных – графиков [16]. Архивация данных в MasterSCADA может осуществляться в собственный файловый архив или в одну из распространённых СУБД (MS SQL, Oracle, Firebird). Для графического представления архивных значений параметров используется модуль трендов (рис. 2.4).

Модуль трендов в MasterSCADA имеет обширную функциональность:

- практически полное управление составом служебных элементов тренда, его внешним видом;
- различные способы навигации по архивным данным, в том числе с использованием закладок, моментов нарушения границ и др.;
- обработка перьев (фильтрация и др.), операции над двумя перьями (вычитание, корреляция и т.п.);

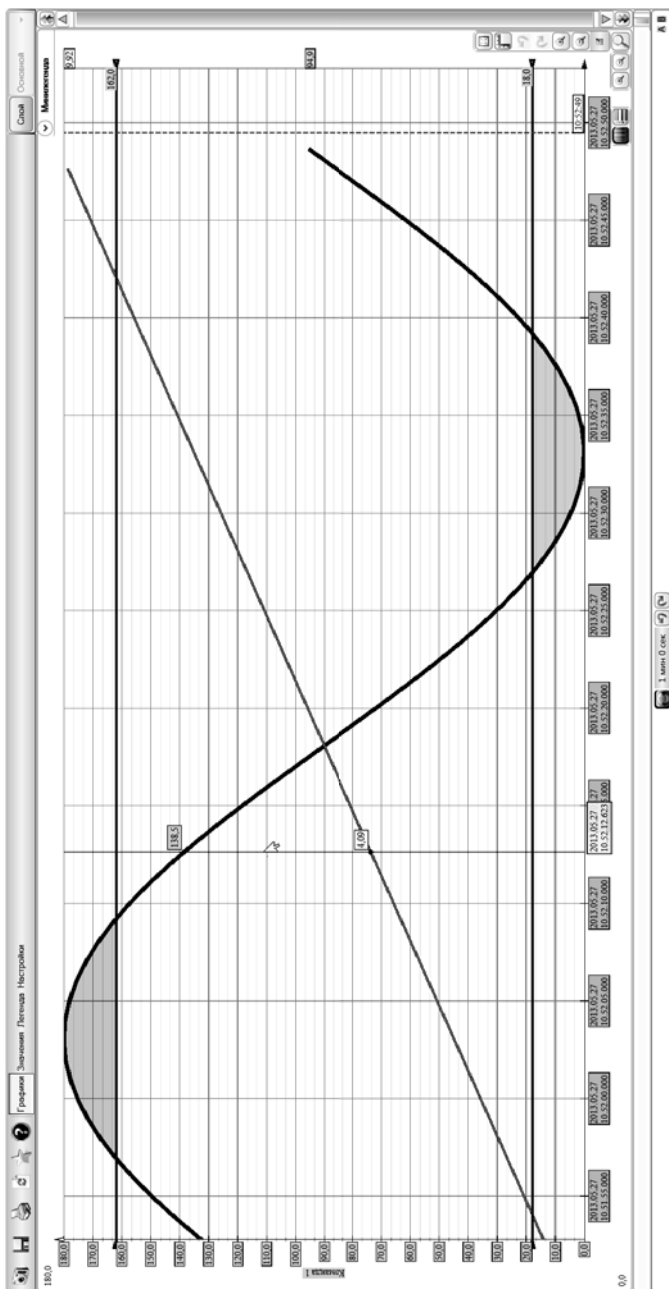


Рис. 2.4. Тренды параметров MasterSCADA

- выделение зон нарушения границ;
- табличное представление параметров;
- экспорт в различные форматы – как в графические (png, bmp, xps, jpg), так и табличные (csv).

Число трендов в проекте или у объекта неограниченно. Тренд можно открывать как окно, документ, или располагать на мнемосхеме как обычный элемент, что позволяет разработчику выбрать наиболее подходящий способ представления графиков.

Добавление переменных на тренд, как и все действия в MasterSCADA, выполняется методом «перетаски и брось» простым перетаскиванием мыши из дерева проекта. Также существует возможность средствами тренда выбрать нужные для отображения переменные из объектов прямо в режиме исполнения.

2.2.4. СООБЩЕНИЯ И ЖУРНАЛЫ MASTERSCADA

В режиме исполнения необходимо оповещать оператора о различных событиях. Важным элементом SCADA-системы является система уведомления оператора о событиях, происходящих на объекте. В MasterSCADA есть системные сообщения (сообщения о нехватке места на диске, отказ OPC-сервера и т.д.), стандартные встроенные сообщения от переменных и функциональных блоков (нарушение контролируемых границ изменения параметров и т.п.), а также пользовательские сообщения, формируемые по любым определённым пользователям в проекте событиям. Количество сообщений в проекте не ограничивается.

Сообщения делятся на разные категории с разным уровнем приоритета, что позволяет наглядно выдавать оператору (диспетчеру) информацию.

Настройка категорий сообщений осуществляется на закладке «Сообщения» свойств дерева системы (рис. 2.5). На этой закладке можно изменить или добавить пользовательские сообщения, приоритет сообщений, указать канал вывода сообщений, сформировать внешний вид сообщения, назначить сообщению при необходимости воспроизводить звуковой файл.

В MasterSCADA сообщению по степени важности назначается один из шести приоритетов: максимальный, очень высокий, высокий, средний, обычный, низкий.

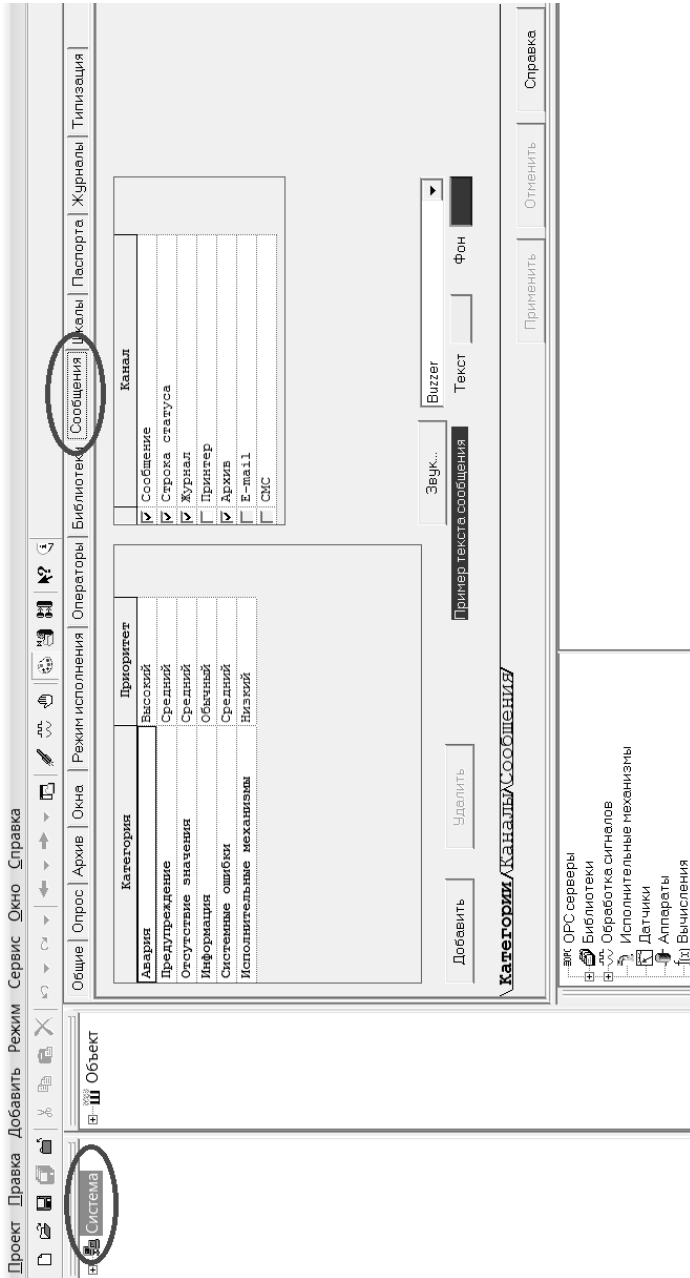


Рис. 2.5. Закладка «категории сообщений» свойств системы

Каждой категории можно назначить канал вывода сообщения. Канал вывода – это способ передачи сообщения оператору или обслуживающему персоналу. В MasterSCADA следующие каналы вывода сообщений (рис. 2.5):

- Сообщение – выглядит как всплывающее окно с текстом;
- Строка статуса – текст сообщения выводится в специальной строке в нижней части окна;
- Журнал – сообщение заносится в специальный журнал, в котором можно осуществлять навигацию по архиву сообщений;
- Принтер – служит для вывода сообщения на печать;
- Архив – сообщение записывается в архив сообщений;
- E-mail – используется для отправки сообщений по электронной почте;
- СМС – используется для отправки СМС-сообщений одному или нескольким абонентам.

Каждой категории можно назначить один или несколько каналов вывода сообщений.

С сообщениями можно работать и через журналы сообщений (рис. 2.6), которых может быть неограниченное количество. Журналы имеют доступ к полному архиву сообщений и могут быть специализированы по категориям, приоритетам или источникам информации так, как это выберет разработчик проекта.

Журнал сообщений имеет несколько столбцов:

- Столбец «Квит.» – в этом столбце можно увидеть квитировано сообщение или нет и при необходимости квитировать его;
- Столбцы «Начало» и «Конец» показывают диапазон времени, когда событие, по которому формируется сообщение, было активным;
- Столбец «Сообщение» – в этом столбце отображается текст сообщения;

Столбцы «Источник» и «Объект» соответственно отображают источник, который сформировал сообщение, и объект, которому принадлежит источник.

Журналы сообщений предоставляют развитые возможности для работы с сообщениями (сортировка сообщений, использование различных типов фильтров).

2.2.5. РЕДАКТОР ОТЧЁТОВ MASTERSCADА

Важным моментом в работе автоматизированных систем управления является документирование процесса – создание отчёта, содержащего информацию о результатах работы за определённый промежуток времени, производстве продукции, расходе материалов, об уставках, значениях, отклонениях параметров от норм и т.д.

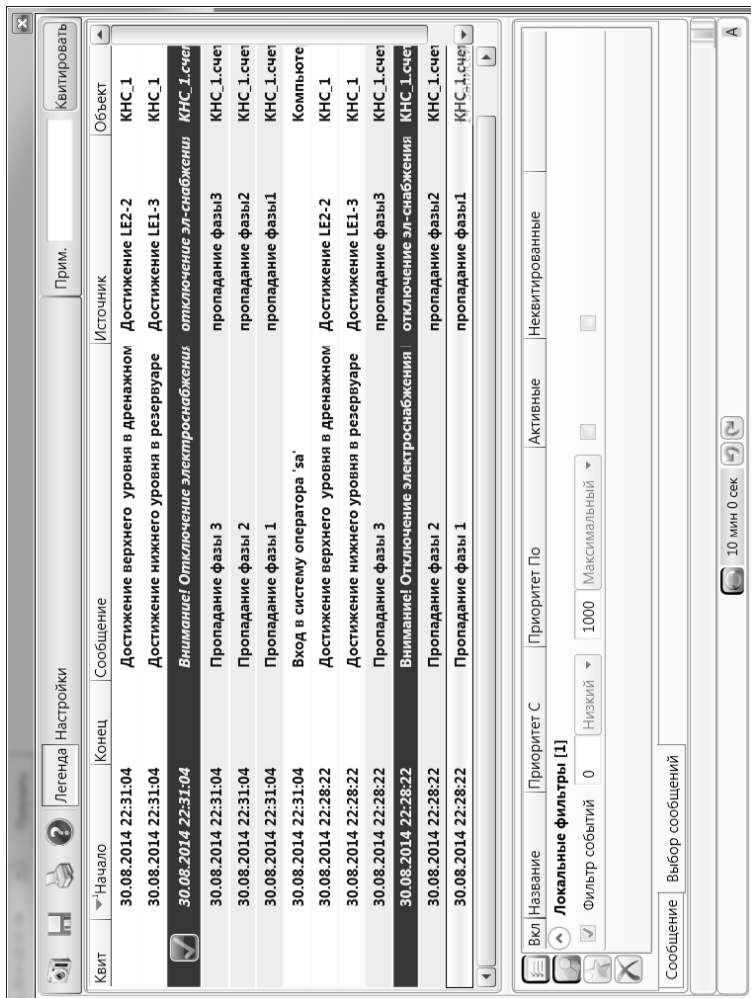


Рис. 2.6. Журнал сообщений

В MasterSCADA есть встроенный редактор отчётов – Master-Report. Редактор отчётов имеет высокую функциональность и позволяет создавать отчёты любых форм [17].

Отчёт состоит из специальных секций – бэндов, которые определяют расположение тех или иных частей отчёта, например, заголовков, таблиц, диаграмм и графиков. Это позволяет создавать отчёты очень сложных форм с точным позиционированием информации на листе.

Отчёты содержат развитые средства получения данных из различных источников: текущих и архивных значений переменных MasterSCADA, выборки из баз данных с помощью SQL-запросов и др.

В отчётах есть возможность обработки полученных данных, как заданием формул с использованием больших библиотек функций (в том числе, специальных видов суммирования: страничного, нарастающим итогом и т.п.), так и с использованием языка сценариев C#.

В отчётах есть возможность включения любой графической информации, включая графики, диаграммы, штрих-коды и т.п.

В редакторе отчётов есть возможность предварительного просмотра создаваемого отчёта – без запуска системы в режим исполнения, что значительно упрощает и ускоряет создание отчёта.

Сформированный в режиме исполнения отчёт можно сохранить в различные форматы – pdf, xls,xlsx, doc, HTML, HTML5, форматы пакета Open Office, графические форматы. Есть возможность как ручного сохранения, так и автоматического – с использованием расписания.

2.3. СРЕДА ИСПОЛНЕНИЯ MASTERSCADА

Компания «ИНСАТ» предлагает гибкую лицензионную политику по реализации систем управления, построенных на базе MasterSCADA. Выбор той или иной версии исполнительной системы MasterSCADA (называется MSRT) и их количества зависит от архитектуры системы, количества рабочих мест, требований к надёжности, способу связи с устройствами ввода/вывода сигналов и контроллерами и т.д. MasterSCADA позволяет строить системы управления с различной конфигурацией: от простой автономной системы до сложных систем с развитой сетевой архитектурой и расширенной функциональностью.

На рисунке 2.7 показана наиболее распространённая конфигурация системы. В такой системе компьютер в проекте всего один. В этом случае используется базовая автономная версия MasterSCADA – MSRT.

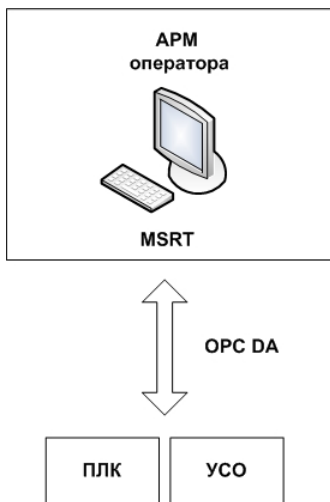


Рис. 2.7. Структура системы управления, построенной на базе автономной версии MasterSCADA – MSRT

В базовой комплектации MSRT может:

- Читать и записывать данные в контроллеры (приборы, модули) через OPC DA-сервер. При этом количество подключённых OPC-серверов не ограничено (ограничено только максимальное количество добавленных OPC-тегов).

- Архивировать полученные данные в файловом архиве – при таком способе архивирования данные записываются в файлы специальной структуры и хранятся в папке проекта.

- Отображать текущие данные на мнемосхемах, архивные данные – на трендах.

- Формировать, архивировать и отображать в журналах различные сообщения – как predeterminedенные системой, так и созданные самим пользователем.

- Формировать пользовательские отчёты, используя штатный редактор отчётов MasterReport.

- Распечатывать журналы, тренды, отчёты, мнемосхемы.

- Экспортировать журналы и тренды в графические форматы (Jpg, png, tiff) и текстовые форматы (csv для трендов, и html для журналов).

- Сохранять отчёты в различные форматы (pdf, HTML5, doc, docs, xls, xlsx, Open Office Documents, jpg, png) как в ручном, так и автоматическом режиме.

Кроме того существует возможность расширить функционал базовой версии исполнительной системы MSRT за счёт приобретения дополнительных модулей:

- HDA-client. При наличии данного модуля MSRT получает возможность получать данные от OPC HDA-серверов. OPC HDA-серверы передают от приборов (контроллеров и приборов коммерческого учёта) архивные данные.

- Модуль связи с базами данных. При наличии данного модуля MSRT получает возможность работать с базами данных – архивировать в неё данные и сообщения, экспортировать данные, а также выполнять хранимые процедуры.

- MSRT-BackUp. При наличии данного модуля MSRT получает возможность экспортировать данные и сообщения в формат mdb – формат базы данных Access. Данная возможность требуется для взаимодействия с различными сторонними программами – программами бухгалтерского учёта, MES-системами, другими SCADA или программами собственной разработки.

- Интернет-сервер. При наличии данной опции появляется возможность просматривать мнемосхемы, тренды, журналы, отчёты через сеть Internet (или локальную сеть) используя обычный браузер.

- Различные библиотеки. Библиотеки содержат набор функциональных блоков для использования в различных отраслях – вентиляция и кондиционирование (HVAC), тепло- и электроэнергетика, коммерческий учёт.

- Дополнительные способы вывода сообщений. В базовой версии сообщения выводятся во всплывающем окне (при необходимости – со звуковым сопровождением), в строку статуса и в журнал сообщений. Дополнительные опции позволяют выводить сообщения через SMS и E-Mail. При этом модуль E-mail допускает использование вложений, что позволяет отправлять ответственным лицам не только сообщения, но и различные файлы – графики, журналы, отчёты, снимки мнемосхем.

- Программирование контроллеров под управлением MasterPLC. Данная опция позволяет программировать контроллеры с открытой архитектурой (OBEH, ICP DAS, Advantec, Мох, Текон) непосредственно в среде MasterSCADA.

В случае если в проекте используется более одного компьютера, то необходимо использовать сетевую версию MasterSCADA – MSRT-Net. При этом конфигурация проекта может быть разной. Например, компьютер оператора и компьютер мастера (рис. 2.8).

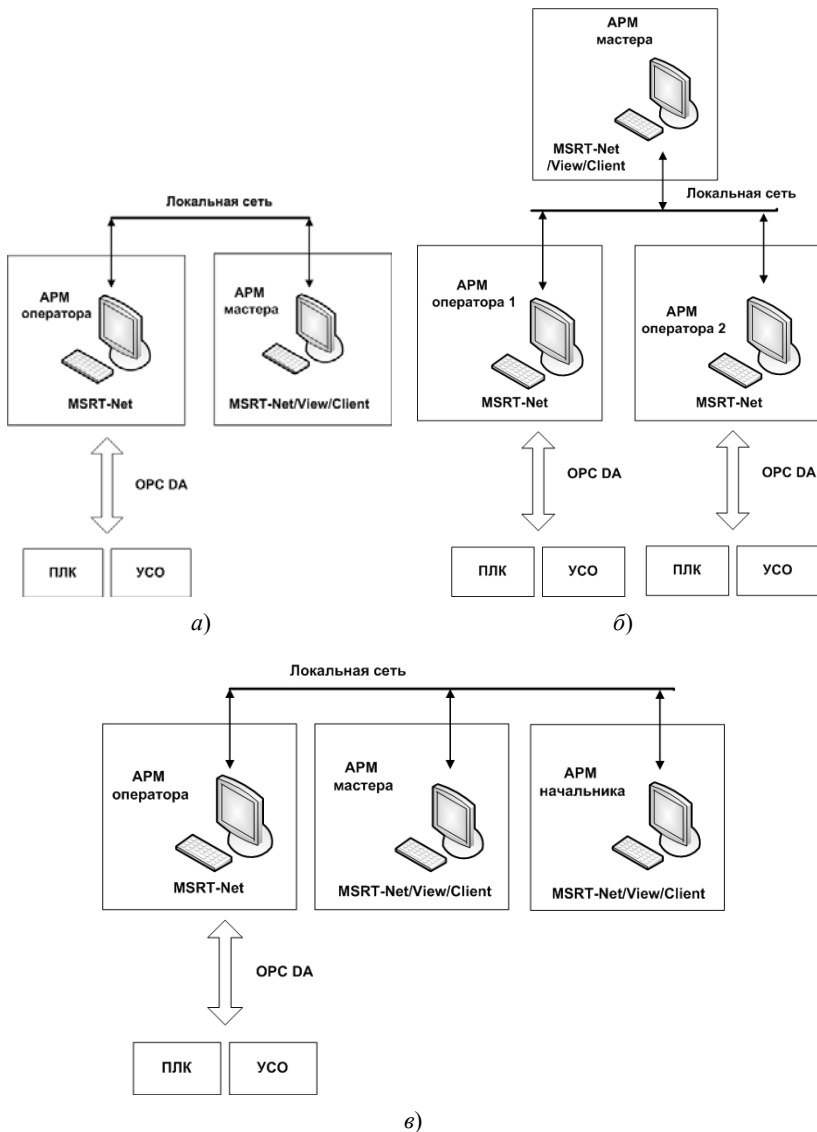


Рис. 2.8. Структуры системы управления, построенной на базе сетевых версий MasterSCADA – MSRT-Net и MSRT-Net/View/Client:

а – с одной станцией оператора (сервером) и одним удалённым клиентом;

б – с несколькими станциями оператора (серверами) и одним клиентом;

в – с одной станцией оператора (сервером) и несколькими клиентами

Количество компьютеров в проекте и конфигурация проекта может быть разной, но во всех случаях необходимо использовать сетевую версию MSRT для каждого компьютера в проекте. Наряду с полнофункциональной сетевой версией MSRT-Net имеются версии с ограничением части функций: сетевые клиенты – MSRT-View и MSRT-Client. Сетевые клиенты предназначены для мониторинга процесса – в них нельзя добавить OPC-серверы, контроллеры, БД-коннекторы, но при этом клиенты имеют неограниченное количество лицензионных связей с другими компьютерами в сети, т.е. сетевые клиенты удобно использовать для рабочих мест, взаимодействующих с основным сервером (сетевой версией MSRT-Net) – для технологов, мастеров, руководителей и других пользователей, не занимающихся оперативным управлением технологическим процессом.

Сетевые клиенты выпускаются в двух версиях:

- MSRT-View – сетевой клиент без функций управления, т.е. с данного компьютера можно просматривать мнемосхемы и тренды, открывать окна. Но послать какую-либо команду другим компьютерам нельзя.

- MSRT-Client – сетевой клиент с функциями управления. Данная версия позволяет вести управление – передавать данные другим компьютерам.

При работе в сетевой версии компьютеры в проекте могут обмениваться данными – т.е. один компьютер может отображать данные с другого, кроме того существует возможность навигации по мнемосхемам и любым документам другого компьютера.

Также возможно изменять значения переменных, принадлежащих другому компьютеру или подключённому к нему оборудованию (посылать команды). При необходимости, можно заблокировать те или иные действия, производимые по сети, используя настройки прав доступа.

В базовой комплектации MSRT-Net может обмениваться данными (текущими, архивными, архивом сообщений) с другими сетевыми версиями в сетевом проекте. Также сетевая версия MSRT-Net имеет те же функции, что и автономная, и также для расширения её функциональности могут быть приобретены дополнительные модули.

На рисунке 2.9 представлена ещё одна разновидность сетевой системы – система с архивным сервером. В такой системе возможно архивирование данных в одну из поддерживаемых СУБД (MS SQL, Oracle, Firebird и др.).

Компьютер, в который добавлен БД-коннектор для связи с базой данных, а также опция MSRT-ARC-Server, называется MAS. Кроме того в MAS включена опция HDA-client – т.е. MAS может опрашивать OPC HDA-серверы (OPC с поддержкой чтения архивов).

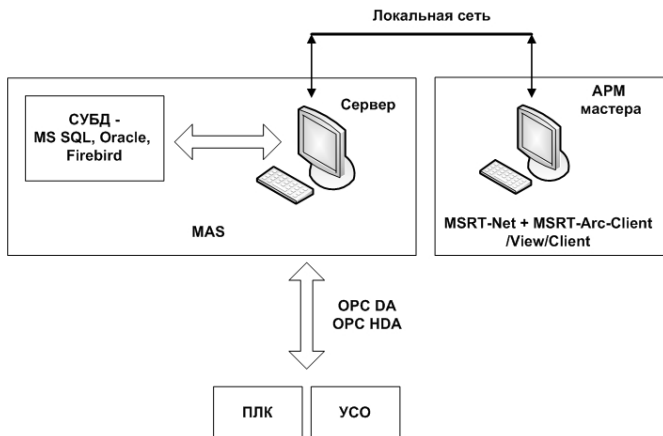


Рис. 2.9. Система управления с выделенным архивным сервером

Для сложных и ответственных производств рекомендуется использовать резервирование. В такой системе происходит резервирование пары компьютеров. При этом возможен опрос одного контроллера обеими частями резервированного комплекта (рис. 2.10).

Также возможно построение систем, в которых осуществляется резервирование как компьютеров, так и контроллеров (рис. 2.11).

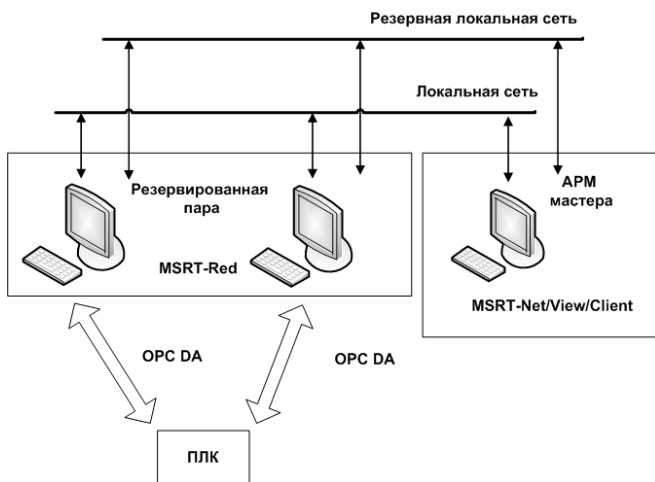


Рис. 2.10. Структура системы управления с резервированием компьютеров

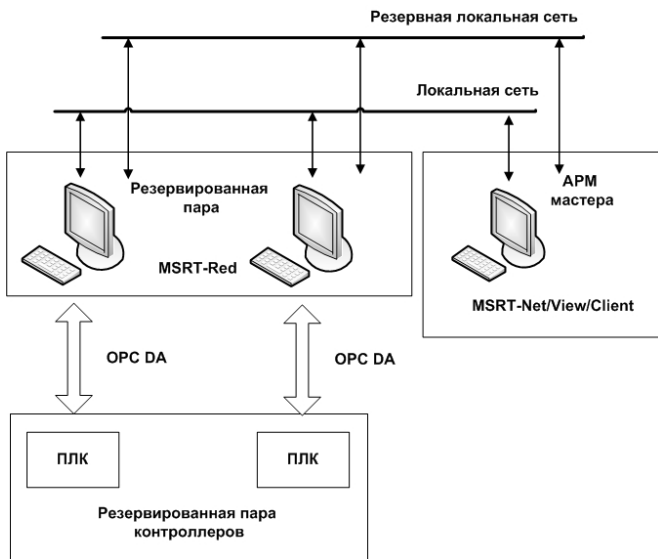


Рис. 2.11. Структура системы управления с резервированием компьютеров и контроллеров

Один из компьютеров является основным, другой – резервным. В случае возникновения отказа (поломки компьютера, отсутствия связи с контроллером, отказа OPC-сервера) происходит переключение с основного на резервный.

2.4. СОЗДАНИЕ ПРОСТЕЙШЕГО ПРОЕКТА

Постановка задачи:

Необходимо построить автоматизированное рабочее место (АРМ) оператора канализационной насосной станций.

Состав имеющегося технологического оборудования:

- приёмный резервуар с датчиком уровня (показания уровня – 0...100%) – 1 шт.;
- сетевые насосы – 3 шт.;
- прибор контроля качества электроэнергии – 1 шт.;
- датчик проникновения и датчик задымления (пожара) с дискретными выходными сигналами;
- щит управления, который содержит программируемый логический контроллер, осуществляющий связь с датчиками и приборами и производящий управление насосами.

Система контроля и управления должна:

- осуществлять контроль уровня в приёмном резервуаре с отображением информации о степени заполнения резервуара в численном и графическом виде, а также осуществлять сигнализацию о достижении уровнем предупредительных и аварийных границ;
- осуществлять контроль качества электроснабжения объекта с отображением напряжения отдельно по каждой фазе, а также сигнализацию в случае нарушения электроснабжения (пропадание фазы или снижение напряжения ниже допустимой величины, например, 180 В);
- отображать состояние работы сетевых насосов;
- предоставлять возможность дистанционного управления насосами (включать/отключать с АРМа);
- осуществлять сигнализацию в случае несанкционированного проникновения или задымления.

Связь АРМ-оператора и программируемого контроллера осуществляется по технологии OPC. Имитацию работы контроллера в учебном проекте можно осуществить с использованием любого OPC-симулятора, например, «Iconics OPC Simulator Server» компании Iconics или «Matricon OPC Simulator» компании Matricon. Эти программы специально используются для отладки программного обеспечения АСУ ТП и могут быть бесплатно скачены с сайтов производителей этих программных продуктов.

Прежде чем приступать к разработке АРМ-оператора, предварительно установим, запустим и сконфигурируем OPC-сервер. Будем использовать «Iconics OPC Simulator Server».

1. Запустим конфигуратор OPC-сервера. В результате появится стартовое окно конфигуратора (рис. 2.12).

2. Создадим новую папку (например, Parameters), в которой будем располагать симулируемые переменные. Для этого в меню **Edit** выбрать пункт **New...**, а далее – **Folder**. В появившемся окне ввести имя папки, например, Parameters.

3. В папке Parameters создадим новую аналоговую переменную Uroven, которая будет имитировать уровень в резервуаре. Для этого при отмеченной папке Parameters выбрать в меню **Edit** выбрать пункт **New...**, а далее – **Data Item**. В появившемся окне (рис. 2.13) в поле Name ввести имя переменной – Uroven, в поле Signal выбрать тип имитирующего сигнала, например, $100 * \text{Triangle}(T/10)$ (треугольная импульсы с амплитудой 100), а в поле Data Type выбрать тип R4 (число типа REAL 4 байта).

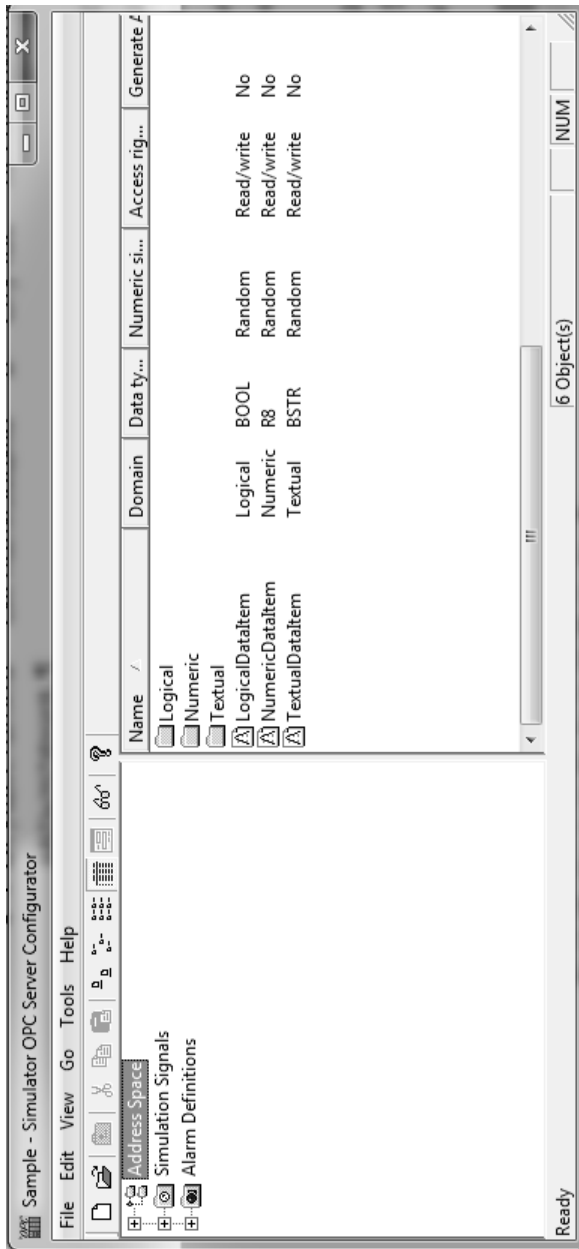


Рис. 2.12. Стартовое окно конфигуратора OPC-сервера Iconics

4. Переменных, имитирующих сигналы о напряжении по фазам в конфигураторе OPC, создавать не будем. В дальнейшем эти сигналы будем изменять вручную непосредственно в режиме работы созданного проекта APM в среде MasterSCADA. Тем самым покажем, что отладку проекта можно осуществлять и без реальной привязки переменных MasterSCADA к OPC-тегам.

5. В папке Parametrs создадим новую логическую переменную Fire, которая будет имитировать возникновение задымления в машинном отделении КНС. Для этого при отмеченной папке Parametrs выбрать в меню **Edit** выбрать пункт **New...**, а далее – **Data Item**. В появившемся окне (рис. 2.14) в поле Name ввести имя переменной – Fire, выбрать область – Logical, тип сигнала (Data Type) – BOOL, а в поле Signal выбрать тип имитирующего сигнала, например, Logical Square(T/10).

6. В папке Parametrs аналогичным образом создадим логические переменные, отвечающие за состояние насосов Н1, Н2 Н3 – Н1_sost, Н2_sost, Н3_sost, при этом можно использовать различные типы имитирующего сигнала, например, Logical Square, Logical Square(T/10), Logical Square(T/100), Random Square.

7. Закрываем конфигуратор OPC-сервера и приступаем к разработке проекта в MasterSCADA.

Напомним общую идеологию разработки проектов в MasterSCADA. Проект в MasterSCADA содержит два дерева. Первое (дерево объектов) описывает иерархию контролируемого объекта и алгоритмы контроля и управления им. Второе (дерево системы) предназначено для описания программно-аппаратных ресурсов системы автоматизации. Проектировщик формирует деревья из элементов – стандартные доступны через контекстное меню, библиотечные элементы (функциональные блоки – ФБ) перетаскиваются из расположенной внизу экрана библиотеки.

Логика проекта описывается связями между конечными элементами деревьев – переменными, входами/выходами ФБ и т.п. Эти связи устанавливаются перетаскиванием источника данных на приёмник или наоборот. На закладках справа от деревьев настраиваются свойства выбранного элемента, а также создаются документы (мнемосхемы, тренды и др.) для «объектов» – основных элементов дерева объектов.

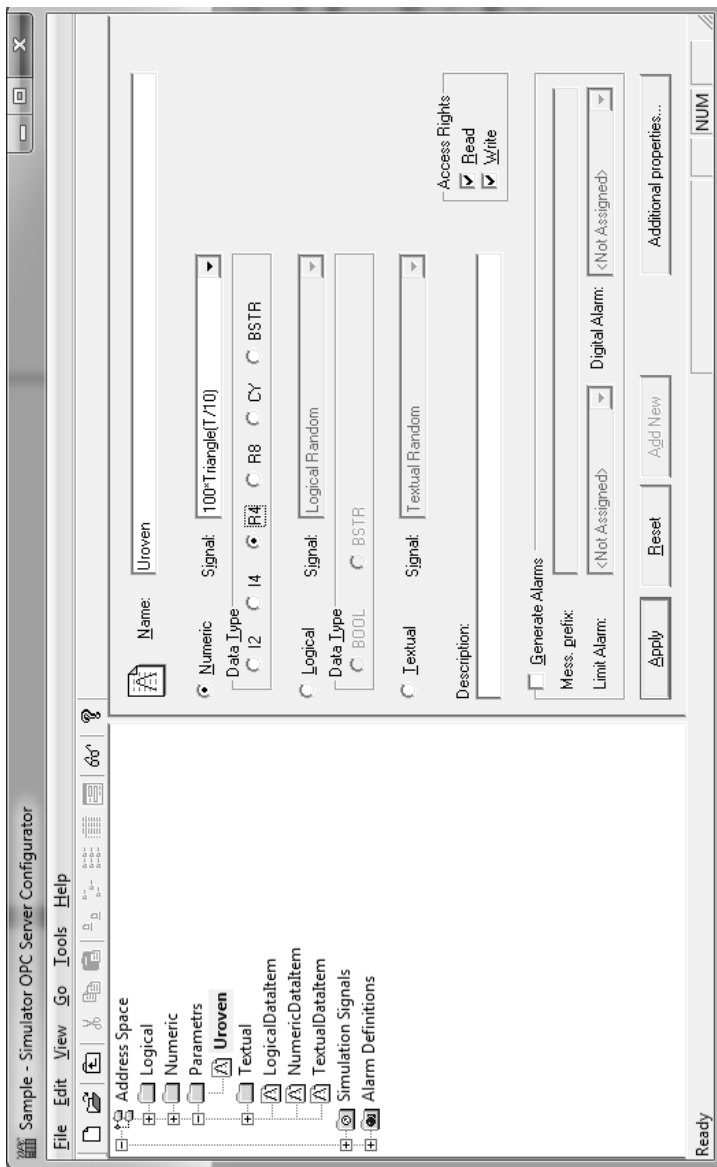


Рис. 2.13. Создание аналоговой переменной Uroven в конфигураторе OPC

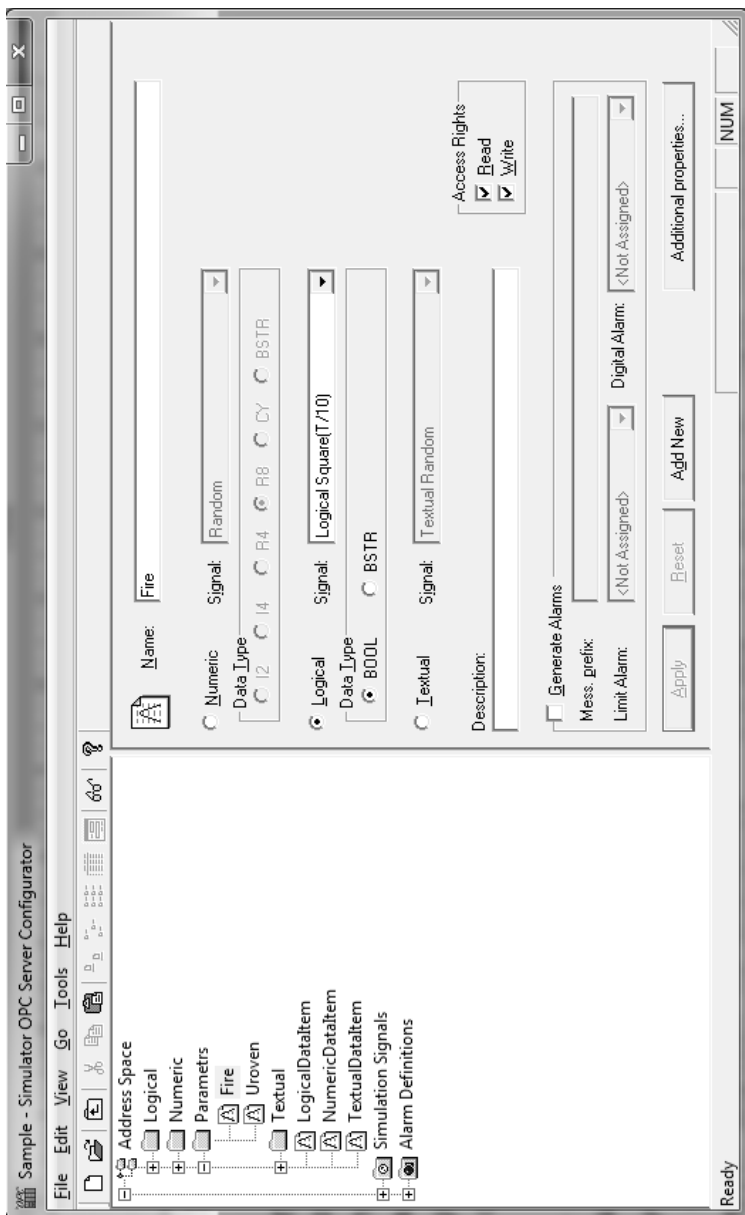


Рис. 2.14. Создание логической переменной Fire в конфигураторе OPC

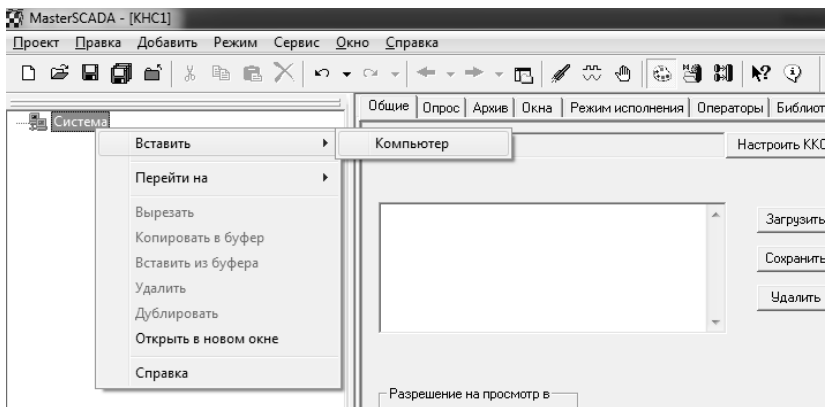



Рис. 2.15. Добавление компьютера в систему

Приступим к разработке проекта в MasterSCADA.

1. Запустим MasterSCADA и создадим новый проект с именем КНС1. Для этого в меню «Проект» выбрать пункт «Создать» или воспользоваться пиктограммой . В открывшемся окне вводим имя проекта КНС1 и отказываемся от установки пароля для доступа к проекту.

2. Осуществим настройку дерева системы. Кликнув правой кнопкой мыши на надписи «Система», вызываем контекстное меню и добавляем компьютер в систему (рис. 2.15).

3. Далее необходимо добавить используемый OPC-сервер. Для этого кликнув правой кнопкой мыши на надписи «Компьютер 1», вызываем контекстное меню, выбираем пункт «Вставить OPC-сервер» и добавляем необходимый OPC-сервер – «Simulator OPC DA Server» (рис. 2.16). Если необходимый OPC-сервер отсутствует в списке, то необходимо произвести поиск доступных OPC-серверов. Для этого необходимо выполнить пункт «Поиск OPC DA серверов ...» контекстного меню.

4. После добавления OPC-сервера добавляем используемые теги (переменные) этого сервера. Для этого также воспользуемся контекстным меню, которое вызывается, кликнув правой кнопкой на имени сервера (рис. 2.17). В открывшемся окне отмечаем добавляемые OPC-переменные.

5. Приступим к настройке дерева объектов. Создадим в дереве объектов MasterSCADA элемент («объект») «КНС», вызвав правой кнопкой мыши контекстное меню и выбрав пункт «Вставить» и далее «Объект».

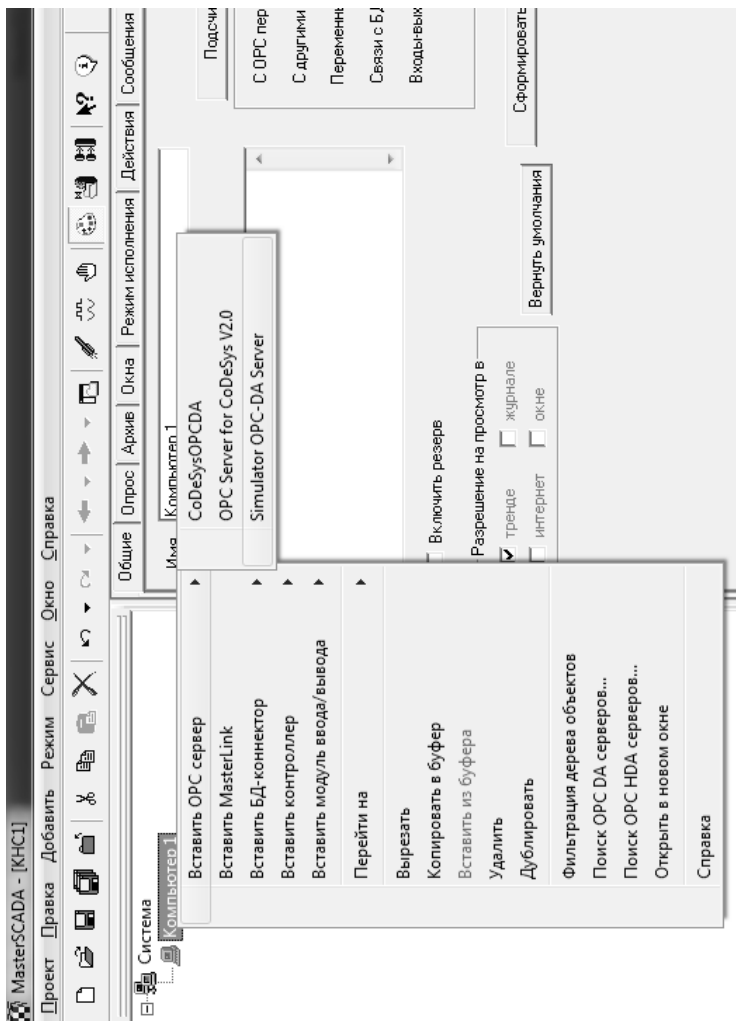


Рис. 2.16. Добавление OPC-сервера в систему

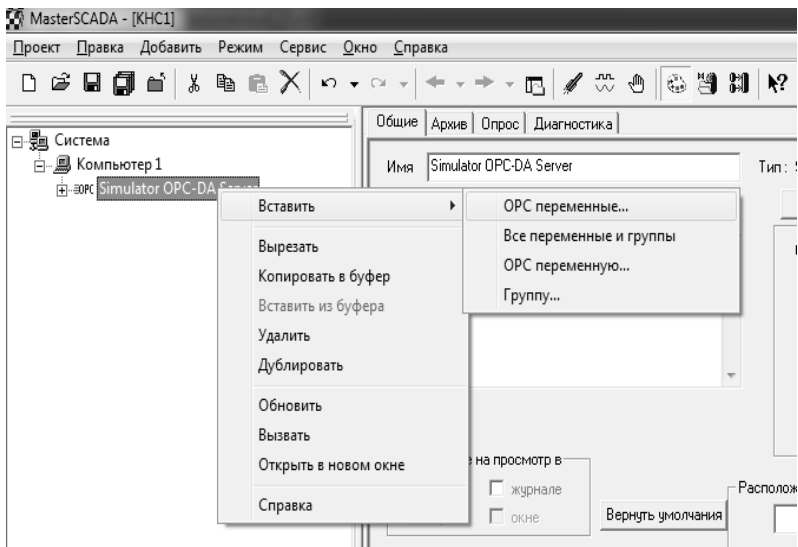


Рис. 2.17. Добавление OPC-переменных

6. Осуществим формирование списка параметров. Вставим в объект «КНС» аналоговые входы: измеряемый параметр – «уровень», напряжения по фазам – «фаза_А», «фаза_В», «фаза_С»; дискретные датчики: «состояние_Н1», «состояние_Н2», «состояние_Н3», «задымление», «проникновение». Эти параметры мы вставим не как переменные, а с помощью библиотечного функционального блока «Повторитель» (рис. 2.18). Такой подход позволяет «централизовать» в одном месте все связи проекта с реальными источниками сигнала, благодаря чему проще и быстрее подключить оборудование на последующих этапах разработки или заменить его тип в дальнейшем. Для каждого ФБ необходимо установить требуемый тип выхода: для аналоговых параметров («уровень», «фаза_А», «фаза_В», «фаза_С») – вещественный (рис. 2.19), для дискретных («состояние_Н1», «состояние_Н2», «состояние_Н3», «задымление», «проникновение») – логический. Затем каждому параметру (выходу ФБ «Повторитель») можно выбрать шкалу контроля границ. В системе имеется ряд predetermined шкал. Если их не хватает, то они задаются на соответствующей закладке корневого узла дерева системы.

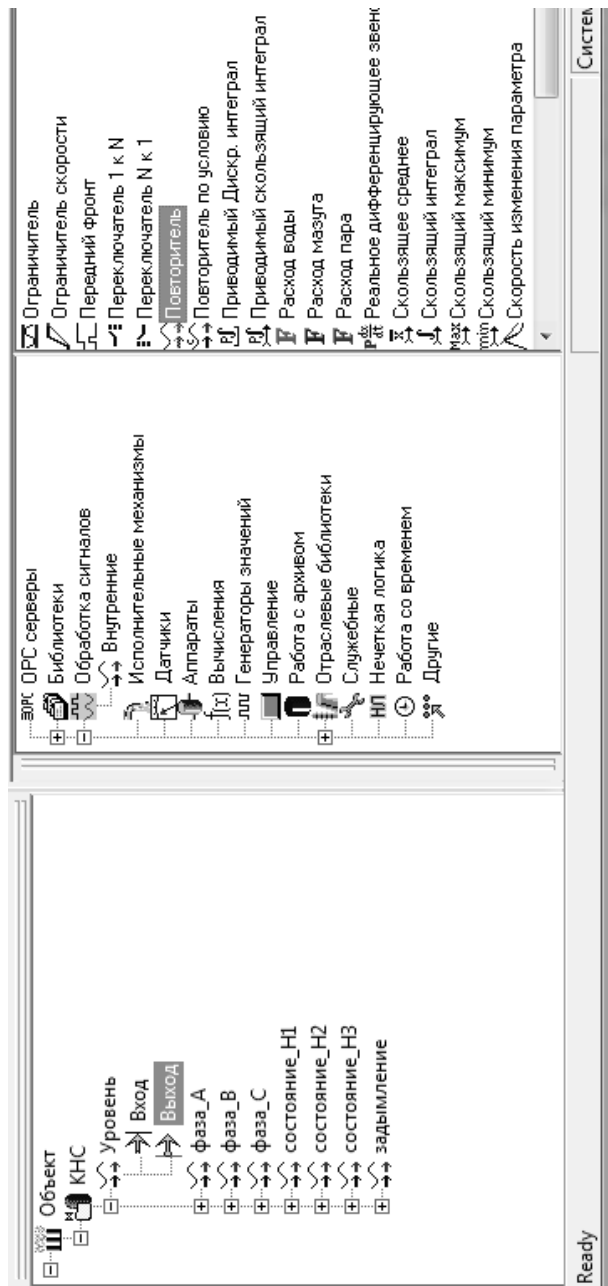


Рис. 2.18. Вставка параметров с помощью библиотечного функционального блока «Повторитель»

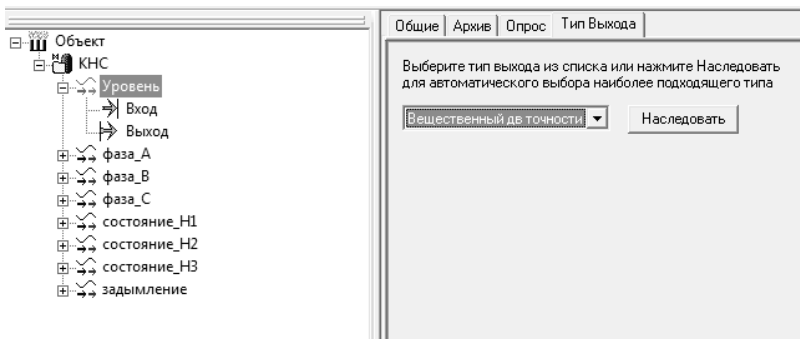


Рис. 2.19. Установка типа выхода для параметра «уровень»

7. Перетащим из палитры функциональных блоков в дерево объектов (в объект «КНС») ФБ «Динамический ИМ» и заменим его название на «насос_H1». Затем произведём настройку этого ФБ (рис. 2.20).

8. Продублируем в дереве объектов через контекстное меню элемент «насос_H1» и сформируем новые элементы «насос_H2» и «насос_H3».

9. Для организации сигнализации в случае нарушения электропитания (пропадание фазы или снижение напряжения ниже допустимой величины, например, 180 В) вставим через контекстное меню новый элемент – событие, нажав правую кнопку мыши на объекте «КНС», далее «Вставить» и «Событие». Переименуем событие на «Нарушение питания». Для редактирования логики работы перейдем на вкладку «Формула» этого события (рис. 2.21). По условию задачи сигнализация о нарушении электроснабжения должна возникать в случае, если напряжение одной из фаз будет менее 180 В. Из дерева объектов в поле «Источник данных» перетаскиваем левой клавишей мыши выходы элементов «фаза_A», «фаза_B» и «фаза_C». Затем записываем условие срабатывания события.

10. Произведём привязку переменных дерева объектов к соответствующим переменным дерева системы. Это осуществляется простым перетаскиванием с помощью левой клавиши мыши. В случае удачной привязки у привязанной переменной появляется вертикальная черта розового цвета (→ Выход).

11. Создадим мнемосхему у объекта «КНС». Для этого сначала открываем закладку «Общие» у объекта «КНС» и в поле «Компьютер» выбираем «Компьютер 1». Затем переходим на закладку «Окна», выбираем из списка тип окна «Мнемосхема», устанавливаем отметку «Стартовая мнемосхема компьютера» и нажимаем кнопку «Создать» (рис. 2.22).

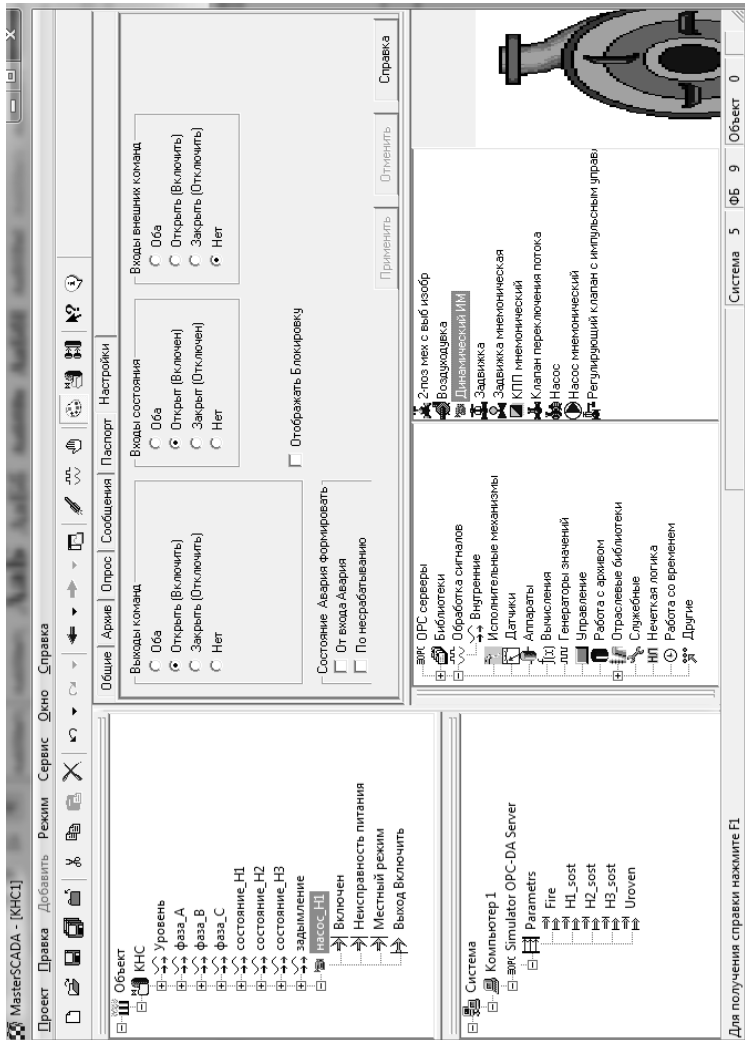


Рис. 2.20. Добавление ФБ из палитры функциональных блоков в дерево объектов

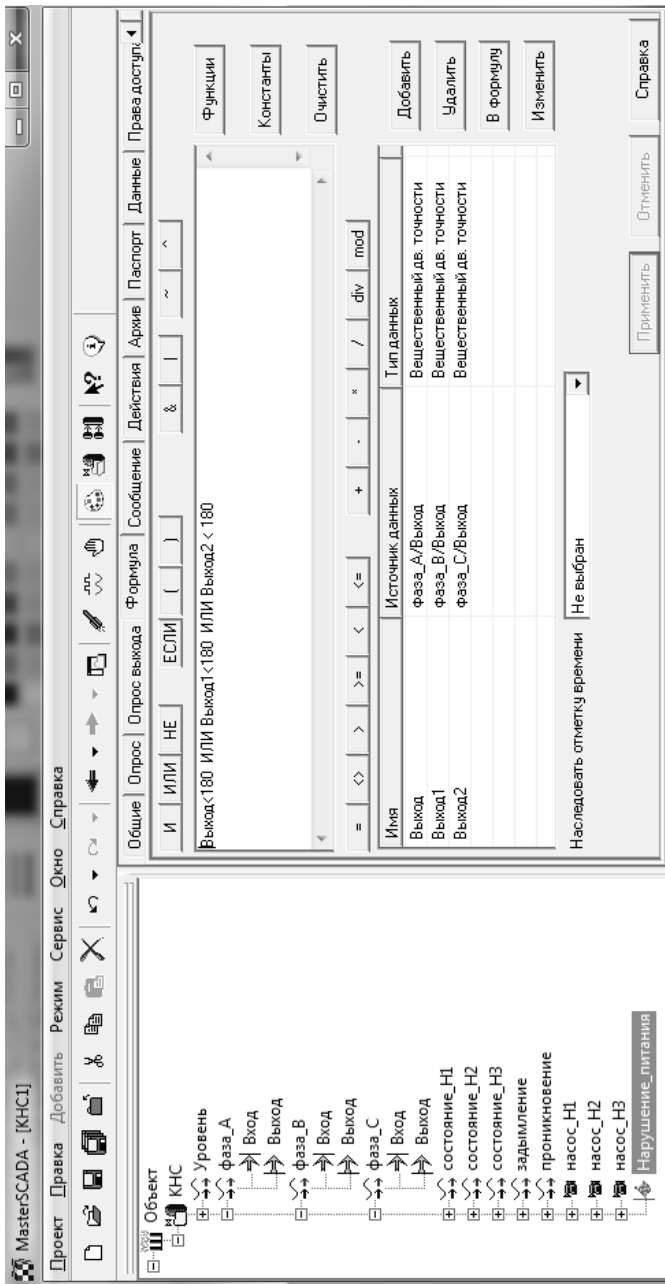


Рис. 2.21. Редактирование события «Нарушение питания»

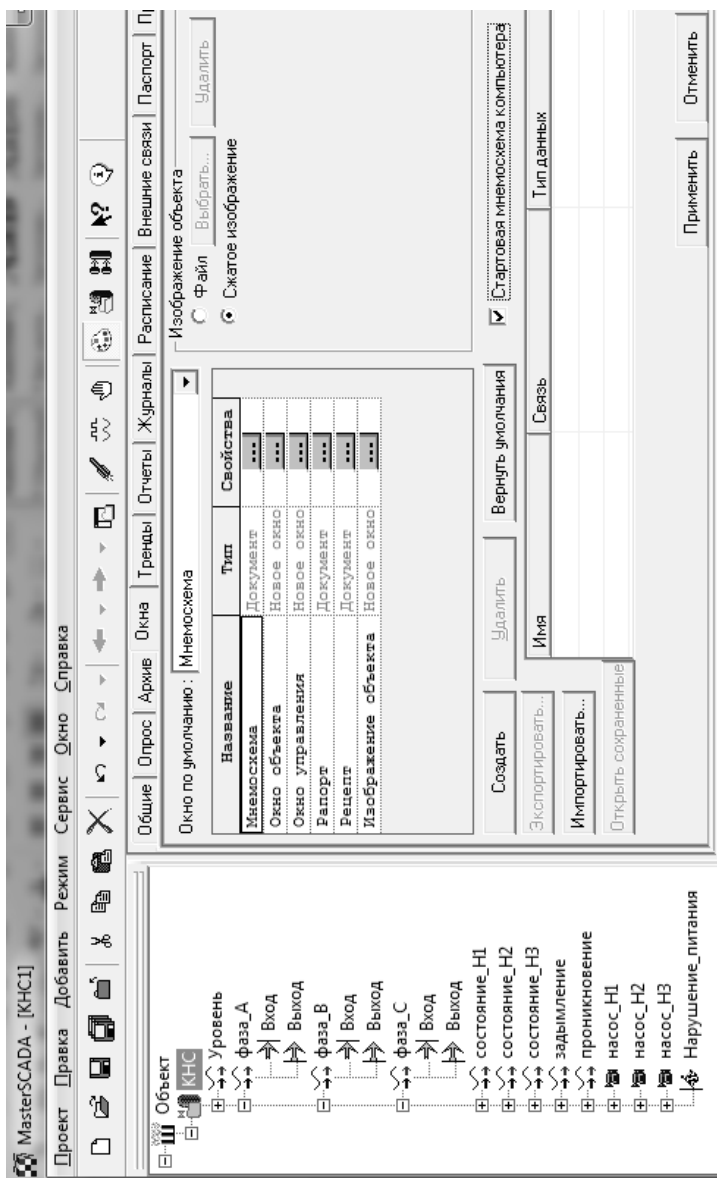


Рис. 3.22. Создание мнемосхемы

12. Приступим к редактированию мнемосхемы. Выберем на палитре графических примитивов в закладке «Объёмные элементы» элемент «Цилиндр» и нарисуем его на поле мнемосхемы. Далее перетащим с помощью мыши из дерева объектов на поле мнемосхемы динамические функциональные блоки «насос_Н1», «насос_Н2» и «насос_Н3». С помощью объёмных графических примитивов «Трубопровод» соединим резервуар и насосы (рис. 3.23). Перетащим на мнемосхему из дерева объектов мгновенное значение уровня (выход повторителя «Уровень») и отредактируем свойства этого элемента (рис. 2.24). MasterSCADA позволяет представлять любые значения как в текстовом виде, так и в виде щитовых приборов (стрелочных, столбиковых, диаграммных). Для этого перетаскивание значений из проекта в мнемосхему нужно делать правой клавишей мыши. Именно в виде стрелочных приборов отображим значения параметров «фаза_А», «фаза_В», «фаза_С». Элементы мнемосхемы, созданные перетаскиванием из дерева объектов, не требуют никакой дополнительной настройки: они уже содержат связь с исходным значением и отображают не только саму цифру или её графическое представление, но и событие её выхода за заданные шкалой контролируемые границы (цветом), недостоверность значения (например, при обрыве связи).

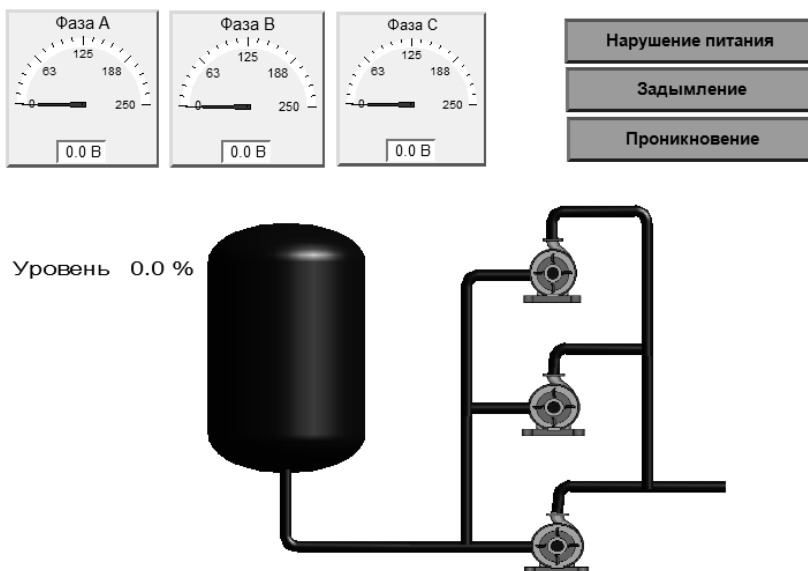


Рис. 2.23. Внешний вид мнемосхемы «КНС1»

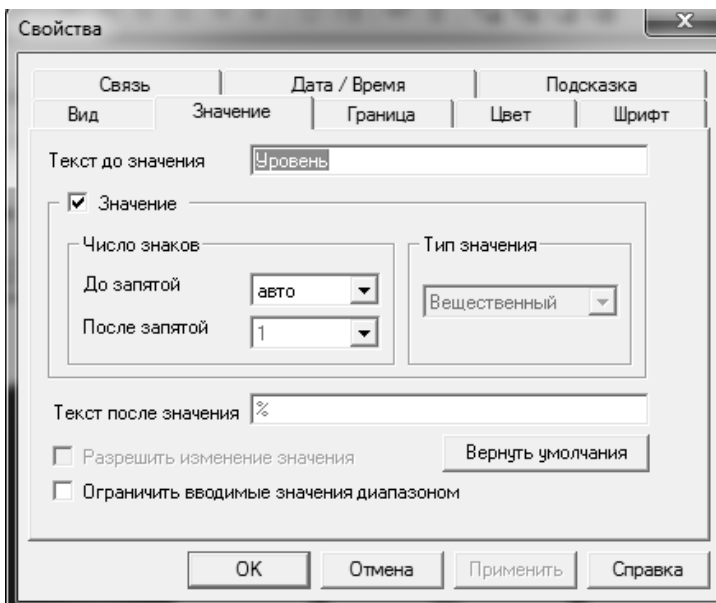



Рис. 2.24. Закладка «Значение» свойств элемента «Уровень» мнемосхемы

13. Создадим элементы сигнализации «Нарушение питания», «Задымление» и «Проникновение». С помощью графических примитивов «Прямоугольник трёхмерный» и «Текст» создадим элементы (см. рис. 2.25), которые будут изменять своё поведение (например, начинать мигать) при возникновении соответствующих событий. Далее произведём динамизацию трёхмерного прямоугольника, находящегося под надписью «Нарушение питания». Для этого выделим прямоугольник и переключимся на закладку «Свойства» (рис. 2.26).

Нажмем на пиктограмму  («Динамизация входных значений») и с помощью левой клавиши перетащим событие «Нарушение питания» из дерева объектов в поле «Мигание». Теперь свойство «Мигание» этого графического примитива стало привязанным к событию «Нарушение питания». В режиме работы при возникновении события «Нарушение питания» прямоугольник начнет мигать с частотой примерно 1 Гц. Аналогичным образом осуществим динамизацию двух оставшихся элементов динамизации, привязав свойства соответствующих прямоугольников к выходам повторителей «Задымление» и «Проникновение».

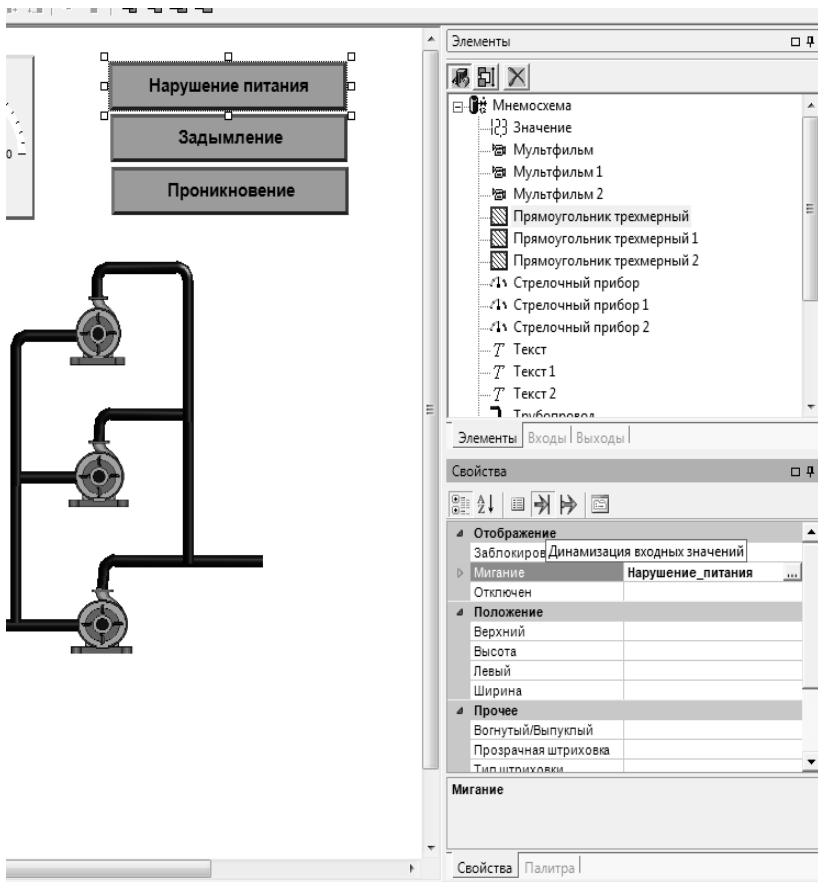



Рис. 2.25. Динамизация входов графического примитива «Прямоугольник трёхмерный»

14. Произведём динамизацию уровня заполнения приёмного резервуара. Для этого аналогично вышеописанному, выделим на мнемосхеме нарисованный резервуар (объёмный элемент «Цилиндр»), переходим на закладку «Свойства» (рис. 2.25), находим свойство «Процент заливки» и перетаскиваем в это поле выход повторителя «Уровень».

15. Сохраним проект, и запустим его на выполнение, нажав на пиктограмму  (Пуск). Параметры аутентификации по умолчанию: имя оператора – sa, пароль – отсутствует.

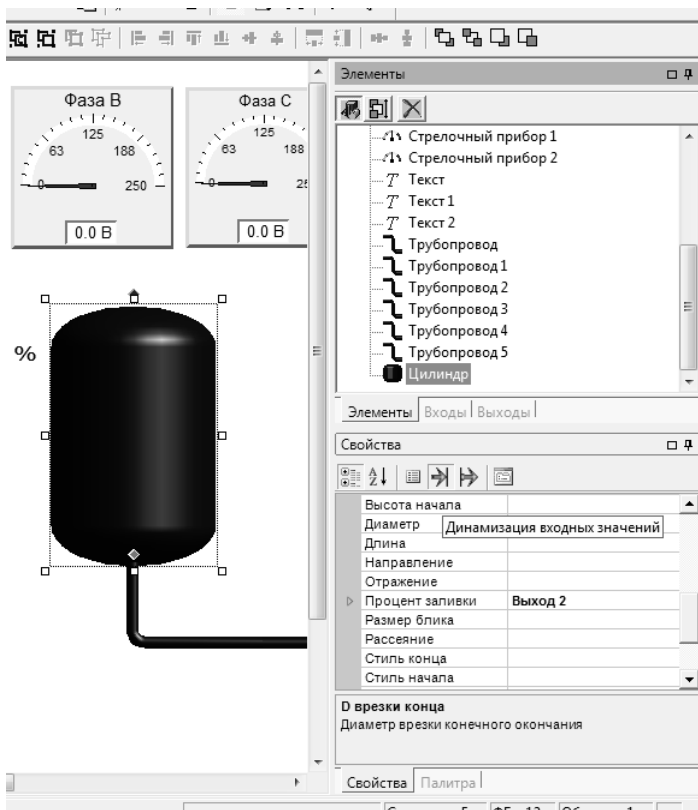


Рис. 2.26. Динамизация входов графического элемента «Цилиндр»

2.5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие основные преимущества имеет пакет MasterSCADA?
2. Что представляет собой объект в MasterSCADA?
3. Какими качествами обладает объект с точки зрения программирования?
4. Приведите примеры свойств объекта и документов объекта.
5. Как проявляется ограничение области видимости для объекта?
6. Какие возможности предоставляет наследование?
7. Какие возможности предоставляет типизация и тиражирование?
8. Какие узлы могут присутствовать в распределённой системе, построенной на базе MasterSCADA? Каково назначение этих узлов?

9. Какие возможности предоставляет MasterSCADA для связи с внешними системами (контроллерами, базами данных и др.)?
10. Какие специальные средства повышения надёжности и живучести разрабатываемых систем предусмотрены в MasterSCADA?
11. Из каких частей состоит среда разработки MasterSCADA?
12. Из каких частей состоит проект MasterSCADA?
13. Что описывает дерево системы?
14. Что описывает дерево объектов?
15. Приведите методику построения дерева системы.
16. На какие типы подразделяются переменные дерева объектов?
17. Что представляет собой «Расчёт»?
18. Что представляет собой «Значение»?
19. Что представляет собой «Команда»?
20. Что представляет собой «Событие»?
21. Что представляют собой функциональные блоки (ФБ) и визуальные функциональные блоки (ВФБ)?
22. Какие группы функциональных блоков присутствуют в палитре?
23. Перечислите базовые документы объектов.
24. Какой основной способ создания мнемосхем?
25. Какие виды элементов проекта могут быть перетащены в мнемосхему?
26. Как могут быть представлены переменные на мнемосхемах?
27. Чем чревата чрезмерная динамизация мнемосхемы?
28. Приведите факторы, влияющие на быстродействие системы, построенной на базе MasterSCADA?
29. Какие возможности имеет модуль трендов в MasterSCADA?
30. Какие виды сообщений предусмотрены в MasterSCADA?
31. Что такое «канал вывода сообщения»?
32. Какие каналы вывода сообщений имеются в MasterSCADA?
33. Приведите структуру базовой автономной версии исполнительной системы MasterSCADA (MSRT).
34. Какие основные возможности базовой комплектации MSRT?
35. Какие имеются возможности по расширению функционала базовой версии исполнительной системы MSRT?
36. Приведите структуры систем управления, построенных на базе сетевых версий MasterSCADA – MSRT-Net и MSRT-Net/View/Client.
37. Приведите структуру системы управления системы с архивным сервером.
38. Приведите структуры систем управления с резервированием.

3. МОДУЛЬНАЯ ИНТЕГРИРОВАННАЯ SCADA КРУГ-2000

3.1. ОБЩИЕ СВЕДЕНИЯ

Первым на отечественном рынке средств автоматизации открытым сертифицированным комплексом, предназначенным для создания АСУ ТП для объектов с сосредоточенными и распределёнными параметрами, является ПТК «КРУГ-2000», разработанный научно-производственной фирмой (НПФ) КРУГ (Контроль Регулирование Управление Гарантии), г. Пенза.

Работа ПТК «КРУГ-2000» поддерживается SCADA-пакетом «КРУГ-2000».

Отличительными чертами SCADA КРУГ-2000 является высокая надёжность, модульность, глубокая интеграция SCADA и среды программирования контроллеров, мощный инструментарий, обеспечивающий удобство создания и отладки проектов пользователя, относительно низкая стоимость по сравнению с западными аналогами [18].

Программное обеспечение (ПО) SCADA КРУГ-2000 имеет мощную базу данных, удобный и простой графический интерфейс, среду разработки программ пользователя КРУГОЛ, модульную среду исполнения и современные средства экспорта/импорта данных.

SCADA КРУГ-2000 имеет модульное построение, как среды разработки, так и систем реального времени (среды исполнения).

Принципиальной особенностью SCADA КРУГ-2000 является предоставляемая Пользователю возможность компоновки и создания своего собственного варианта среды исполнения.

Выбор конфигураций модулей среды исполнения позволяет построить АСУ ТП различной архитектуры:

- локальная система на одном компьютере;
- система, построенная по технологии клиент-сервер;
- иерархическая распределённая система управления многими объектами, интегрированная в сеть предприятия.

Модульный подход к построению автоматизированных систем позволяет использовать только те программные компоненты SCADA КРУГ-2000, которые необходимы для решения поставленных задач. Таким образом, пользователь оплачивает только необходимый ему сейчас функционал, а дополнительные возможности легко наращиваются в ходе дальнейшей эксплуатации.

Модульная структура SCADA КРУГ-2000 минимизирует стоимость, повышает надёжность АСУ ТП.

Пакет «КРУГ-2000» является открытой SCADA-программой. Открытость SCADA КРУГ-2000 достигается следующим:

- поддержкой международных стандартов (Fast Ethernet, RS-485, RS-232, CAN, DeviceNet, TCP/IP, UDP, Modbus, IEC 61131-3 и других);
- использованием международных стандартов для экспорта/импорта данных (OPC, SQL, XML, COM, OLE DB, ODBC).

Для получения данных с нижнего уровня используются:

- встроенная бесплатная библиотека драйверов верхнего уровня;
- библиотека драйверов нижнего уровня;
- комплект средств разработки (SDK) для разработки пользовательских драйверов;
- OPC-клиент, обеспечивающий информационный обмен в соответствии со спецификациями OPC DA и OPC HDA;
- внутренний высокоскоростной канал связи для информационного обмена с контроллерами под управлением системы реального времени контроллеров (СРБК);
- средства импорта данных (из формата XML, XLS, TXT и т.п.).

SCADA КРУГ-2000 позволяет организовать обмен данными с различными устройствами, программами и системами. Для этого используются такие механизмы, как:

- обмен информацией со сторонними приложениями (через DLL, API, OLE, COM, ODBC);
- OPC-сервер SCADA КРУГ-2000, поддерживающий спецификации OPC DA и OPC HDA. Для доступа к данным также возможно использование OPC-клиента SCADA КРУГ-2000;
- интерфейс программирования приложений (API) для доступа к базе данных (БД) реального времени и доступа к архивам;
- средства конвертации и автоматической выгрузки данных в формате XML, Access, Excel и ASCII;
- WideTrack-коннектор, обеспечивающий сбор данных из SCADA КРУГ-2000 в сервер консолидации технологических данных WideTrack. WideTrack позволяет создавать иерархические системы сбора, хранения и обработки информации для крупных компаний и территориально распределённых предприятий;
- библиотека драйверов верхнего уровня;
- SDK для разработки пользовательских драйверов.

Таким образом, SCADA КРУГ-2000 может интегрироваться в единое информационное пространство управления производством.

Пакет «КРУГ-2000» позволяет подключать устройства нижнего уровня управления такие, как контроллеры TREI, ADAM, регулирующие микропроцессорные контроллеры РЕМИКОНТ-Р131, КОНТРАСТ КР-300, КОНТРАСТ КР-300И, КОНТРАСТ КР-300Ш, МПСУ, логические микропроцессорные контроллеры ЛОМИКОНТ, а так же другие УСО, построенные в соответствии с открытыми международными стандартами.

Функции пакета «КРУГ-2000» соответствуют типовому набору информационных и управляющих функций SCADA-программы:

- контроль технологических параметров;
- обнаружение, сигнализация и регистрация отклонений параметров от установленных границ;
- управление регуляторами и дискретными исполнительными механизмами непосредственно с персонального компьютера;
- выполнение функций автоматического регулирования и дистанционного управления;
- блокировки и защиты;
- контроль и регистрация срабатывания блокировок и защит;
- ручной ввод данных;
- архивирование предыстории параметров;
- формирование и выдача данных персоналу;
- формирование и печать печатных документов;
- выполнение вычислительных задач;
- самодиагностика технических и программных средств;
- оперативная настройка;
- конфигурация программного обеспечения;
- передача данных в другие системы;
- приём данных из других систем.

Основные структурные компоненты ПО SCADA КРУГ-2000 показаны на рис. 3.1 [19].

Главными функциями Среды разработки являются следующие [19, 20]:

- конфигурирование создаваемой системы;
- создание и верификация базы данных реального времени;
- разработка графического интерфейса Пользователя (графической базы данных);
- программирование и отладка программ Пользователя;
- выбор и настройка алгоритмов управления;
- имитация работы созданной системы контроля и управления.

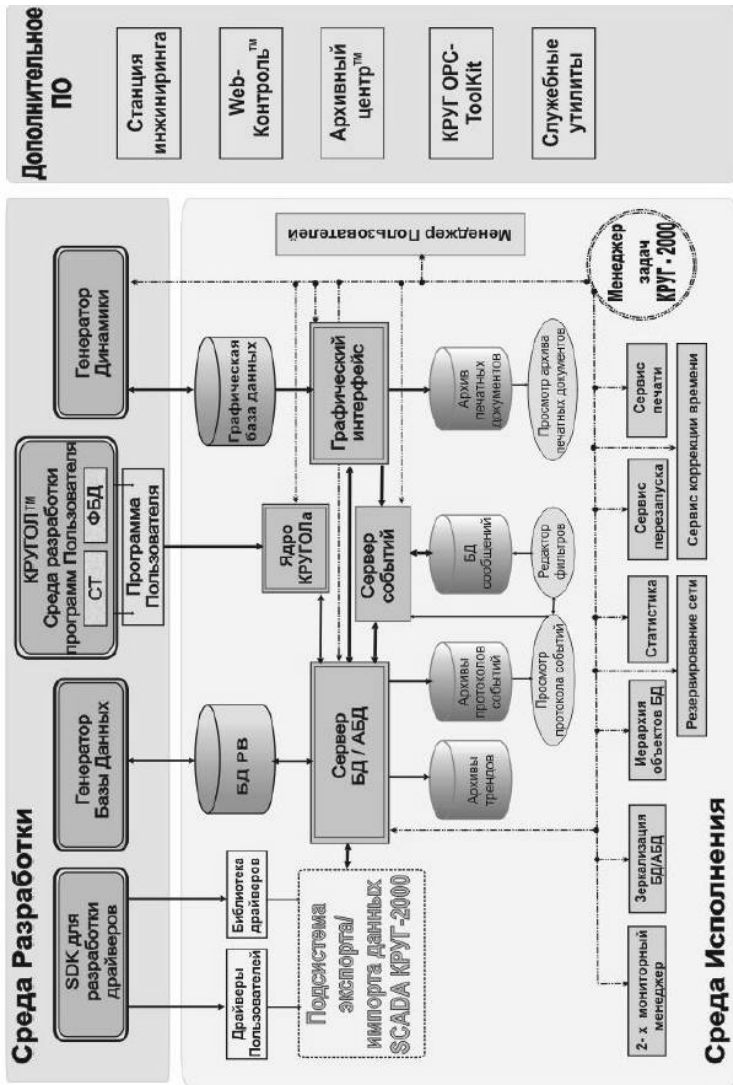


Рис. 3.1. Структура пакета программ системы КРУГ-2000

Главные компоненты среды разработки:

- генератор базы данных (БД);
- генератор динамики (ГД);
- интегрированная среда разработки программ Пользователя

КРУГОЛ.

Генератор базы данных – программное средство для создания и верификации БД реального времени и конфигурирования системы.

Конфигурирование системы включает определение характеристик следующих объектов:

- переменных АСУ ТП;
- УСО и каналов связи;
- абонентов и адаптеров ПТК;
- программ Пользователя;
- принтеров и отчётов и др.

Генератор БД позволяет настроить параметры:

- резервирования;
- администрирования доступа;
- коррекции системного времени;
- объединения и обработки переменных базы данных (БД),

сгруппированных в соответствии со структурой технологического процесса.

Генератор динамики – программный компонент SCADA КРУГ-2000, предназначенный для создания объектно-ориентированного графического интерфейса Пользователя АСУ ТП и генерации отчётов.

Объектами ГД являются:

• мнемосхемы (графические примитивы, виртуальные приборы, тренды, анимация и другие объекты);

- рабочие столы;
- переходы;
- библиотеки изображений и шаблонов.

Язык сценариев (на базе VBScript) предоставляет Пользователю следующие возможности разработки графического интерфейса:

- автоматизация работы операторов:
 - автоматический (по условию) вызов мнемосхем;
 - выдача советов оператору;
 - контроль выполнения советов и подсказок.
- создание интеллектуальных тренажеров и обучающих проектов;
- анимация графических объектов мнемосхем.

КРУГОЛ представляет собой интегрированную среду разработки (ИСР) программ Пользователя. Основными компонентами ИСР КРУГОЛ являются:

- редактор языка структурированного текста (ST);
- редактор языка Функциональных Блочных Диаграмм (FBD);
- компилятор;
- отладчик;
- библиотека алгоритмов управления и обработки данных.

Главные принципы программирования в ИСР КРУГОЛ заключаются в следующем:

- в одну среду разработки объединены средства для программирования на языках FBD и ST;
- программы КРУГОЛ могут использовать общие исходные схемы FBD и тексты ST как в одном проекте, так и в одной программе;
- отладка программ, как для станций верхнего уровня, так и для контроллеров в одной среде;
- расширенны функциональные возможности FBD, в частности добавлены блоки условия и цикла;
- возможность включать функции Пользователя (языки C/C++/Delphi Pascal/...) в библиотеку функций КРУГОЛ для их выполнения на различных платформах (Windows, Linux, QNX) – компонент «Библиотекарь КРУГОЛ»;
- один проект – и для станции оператора и для контроллера;
- один проект для различных платформ (Windows, Linux, QNX).

Основными платформами, на базе которых функционируют программы КРУГОЛ, являются на нижнем уровне – Linux, QNX (контроллеры), на верхнем уровне – Windows (станции оператора).

Компоновка программного комплекса среды исполнения осуществляется из определённого набора исполняемых модулей. Каждый исполняемый модуль отвечает за реализацию определённых функций. Перечень модулей и краткое описание выполняемых ими функций приведён в табл. 3.1, 3.2 [18 – 21].

В процессе эксплуатации у Пользователя может появиться потребность включить в состав программного комплекса новые модули. В пакете КРУГ-2000 предусмотрен простой и удобный механизм наращивания, как функций, так и количества точек ввода/вывода.

Разработана Среда Исполнения для PC-совместимых контроллеров, которая функционирует под управлением операционных систем QNX, LINUX, DOS.

3.1. Основные модули среды исполнения

Название модуля	Основные функции
Сервер БД	Обеспечивает выполнение функций обработки переменных БД, ведения оперативных трендов, программного перезапуска, обработку событий, статистики, ядра технологического языка КРУГОЛ, OPC-сервера, драйвера ODBC
Сервер архивной БД (АБД)	Обеспечивает функции обработки и хранения архивов трендов, протокола событий, печатных документов, OPC-сервера, драйвера ODBC
Просмотр архивов	Модуль для комплексов, работающих с архивными данными. Обеспечивает доступ к архивам трендов и к архивам протоколов событий
Сервер ввода/вывода	Обеспечивает организацию связи различных УСО со SCADA КРУГ-2000 через подключаемые драйверы
Сервер событий	Формирует сообщения из событий, отобранных по заданному фильтру, и передает их для визуализации своим клиентам – Программе просмотра протокола событий или динамическому элементу «Протокол событий» Графического интерфейса
Межсерверный обмен	Обеспечивает обмен информацией между серверами SCADA КРУГ-2000. Обмен производится паспортами выбранных переменных и связанными с ними событиями
Многосерверный доступ	Отображение обобщённой звуковой сигнализации с заданных серверов БД. Смена графического проекта и подключение клиента к другому серверу БД по команде из графического интерфейса
Графический интерфейс	<p>Графический интерфейс оператора. Варианты исполнения: с управлением/ без управления.</p> <p>Под управлением понимается возможность оператора с помощью динамических элементов (кнопка, поле ввода и т.п.) внесение изменений в работу АСУ ТП (например, изменение задания или коэффициентов настройки регуляторов, включение/выключение исполнительных механизмов, изменение границ сигнализации, управление каналами связи и резервированием и т.д.).</p> <p>В варианте без управления пользователь не может вносить какие-либо изменения в ход технологического процесса, кроме квитирования световой и звуковой сигнализации</p>

Продолжение табл. 3.1

Название модуля	Основные функции
Зеркализация БД	«Зеркализация» (периодическое резервное копирование в режиме on-line) данных из основного в резервный сервер. Сеть, по которой осуществляется «зеркализация», назначается Пользователем на этапе конфигурации системы контроля и управления. Основной и резервный серверы БД устанавливаются на двух разных АРМ. «Зеркализуемая» информация: оперативная БД, тренды, печатные документы, протокол событий
Зеркализация АБД	Аналогично «зеркализации» БД. Кроме того, «зеркалируются» архивы исторических трендов, архивы печатных документов и архивы протоколов событий
Автоматический перезапуск аппаратный	Используется при установке в компьютер (АРМ) специальной платы автоматического перезапуска. Совместно с платой обеспечивается автоматический перезапуск операционной системы в случаях «зависания» и «зацикливания»
Сервис печати	Обеспечивает функции автоматической выдачи отчётных документов и протокола событий на устройства печати
Менеджер Пользователей	Ограничение доступа персонала к данным и функциям АСУ ТП. Определение функций доступа и групп Пользователей осуществляется с помощью Генератора базы данных. Формирование списка Пользователей осуществляется с помощью модуля «Менеджер Пользователей»
Файл-обмен	Обеспечивает приём/передачу файлов данных из/в SCADA КРУГ-2000. Данными для приёма/передачи могут быть текущие значения переменных, атрибуты (параметры настройки) переменных. Обмен файлами данных осуществляется как по инициативе внешней системы, так и по инициативе SCADA КРУГ-2000. Данные передаются в двоичном формате
Резервирование сети	Обеспечивает выполнение следующих функций резервирования: сеть контроллер – сервер БД/АБД; сеть сервер БД/АБД – клиенты сервера БД/АБД (в том числе при резервировании серверов и клиентов)

Продолжение табл. 3.1

Название модуля	Основные функции
Статистика	Позволяет провести выборку переменных для любого временного интервала, за который имеются тренды, и применить библиотечные функции статистической обработки. Возможность работы с архивными данными. Полученный результат может быть использован в графическом интерфейсе и при формировании отчётных документов
Иерархия объектов БД	Создание иерархической структуры данных РВ в соответствии с конструкторской, технологической или иной группировкой параметров. Формирование обобщённых признаков сигнализации и выборок. Поддерживает существующие системы кодировок и позволяет создавать собственные
Библиотека драйверов верхнего уровня	Бесплатная библиотека драйверов для различных контроллеров, модулей ввода/вывода и других цифровых приборов
SDK для разработки пользовательских драйверов	Инструментальное программное обеспечение для разработки и выполнения драйверов пользовательских УСО
OPC-сервер SCADA КРУГ-2000 спецификация DA +HDA	Модуль предназначен для передачи оперативных и архивных данных из SCADA КРУГ-2000 в системы сторонних производителей согласно стандартам OPC DA и OPC HDA
ODBC-драйвер	Открытый интерфейс доступа к БД SCADA КРУГ-2000 представляет собой библиотеку функций, которая позволяет прикладной программе обращаться к базе данных SCADA КРУГ-2000, используя структурированный язык запросов SQL

3.2. Дополнительное программное обеспечение

Название модуля	Основные функции
Библиотека функций языка КРУГОЛ	Включает технологические функции, логические и математические функции, функции учёта ресурсов. Функции из библиотеки могут использоваться как в программах для верхнего уровня, так и в программах на языке КРУГОЛ, разработанных для контроллеров под управлением СРВК

Продолжение табл. 3.2

Название модуля	Основные функции
WEB-контроль для SCADA КРУГ-2000	Обеспечивает просмотр текущей информации в виде мнемосхем, печатных документов, трендов, протокола событий на любом персональном компьютере посредством MS Internet Explorer. Взаимодействует с серверами БД и АБД. Поддержка списка «любимых» кадров. Защита от несанкционированного доступа. Возможна работа как в режиме без функций управления, так и с функциями управления
Универсальный конвертер данных	Конвертирует данные (в том числе печатные документы, тренды, протоколы сообщений) из формата КРУГ в формат MS Excel и XML. Позволяет производить выборку по параметрам и по времени
Архивный центр	Предназначен для сбора архивных и других данных от различных АСУ ТП, построенных на базе пакета КРУГ-2000 для Windows, с дальнейшей возможностью их просмотра. Имеет собственный протокол событий, цветовую и звуковую сигнализацию, обладает возможностью дублированного копирования данных. Для просмотра архивов на компьютере Архивного центра необходимо иметь либо Среду Разработки (поставляется бесплатно), либо Универсальный конвертер данных
Модуль модемных каналов связи	Позволяет любым драйверам SCADA КРУГ-2000 и OPC-серверам, разработанным НПФ «КРУГ», использовать GSM- и GPRS-каналы сотовых сетей, линии коммутируемой телефонной сети общего пользования и другие модемные каналы связи
Станция инжиниринга	Конфигурирование и настройка контроллеров; диагностика работы контроллеров в режиме реального времени. Удалённое программирование контроллеров под управлением СРБК. Кроме того, станция инжиниринга позволяет производить диагностику каналов связи, а также выполнять ряд дополнительных сервисных функций

Последовательность разработки АСУ ТП с помощью программных средств SCADA «КРУГ-2000» состоит из следующих этапов [21].

Этап 1. Сбор исходных данных для создания АСУ ТП.

Используются технологическая, функциональная, структурная схемы АСУ ТП. Описание технологических параметров (переменных) рассматриваемого ТП.

Этап 2. Генерация АСУ ТП. Генерация БД.

Осуществляется описание характеристик, таких как: абоненты, каналы, переменные, печатные документы.

Производится настройка параметров: администрирование, печатные документы, самописцы.

Этап 3. Генерация АСУ ТП. Генерация Графической базы данных.

Осуществляется разработка структуры видеок кадров: мнемосхемы, тренды, печатные документы.

Создаются: рабочий стол, переходы, статические и динамические объекты, сигнализация на мнемосхему.

Этап 4. Разработка программ пользователя.

Разрабатываются алгоритмы. Производится написание и отладка программ.

Этап 5. Задание прав доступа.

Определение прав доступа к функциям и данным для персонала АСУ ТП.

Этап 6. Конфигурирование станции оператора и запуск АСУ ТП.

Конфигурирование списка процессов и путей доступа к базе данных и графическому проекту.

Следует отметить, что процесс разработки АСУ ТП с помощью программных средств SCADA «КРУГ-2000» не всегда осуществляется в строгой последовательности, представленной выше, и может иметь итеративный характер.

3.2. ГЕНЕРАТОР БАЗЫ ДАННЫХ СИСТЕМЫ КРУГ-2000

Ядром БД реального времени является файл `db_common.dat`, в котором хранится вся информация о системе контроля и управления. База данных реального времени создаётся генератором базы данных при вызове функции «Сохранить». Создаваемые при этом файлы будут записаны в указанную пользователем директорию.

Запуск Генератора БД системы Круг-2000 осуществляется выбором – «ПУСК / Программы / Система КРУГ-2000 / Среда разработки / Генератор базы данных», при этом необходимо выбрать открываемую БД или создать новую, указав при этом каталог расположения новой базы данных и её название (рис. 3.2).

Восстановление файла БД в формате MS ACCESS в случае его порчи, удаления или неправильных действий пользователя возможно из файла базы данных `db_common.dat`. Перед восстановлением необходимо создать новую базу данных, вызвать функцию «Загрузить» и выбрать директорию, в которой находится файл `db_common.dat`, из которого будет восстанавливаться БД (рис. 3.3).

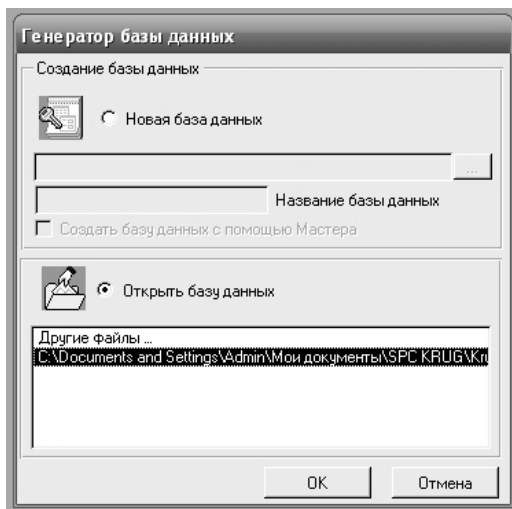


Рис. 3.2. Меню создания (открытия) БД системы Круг-2000

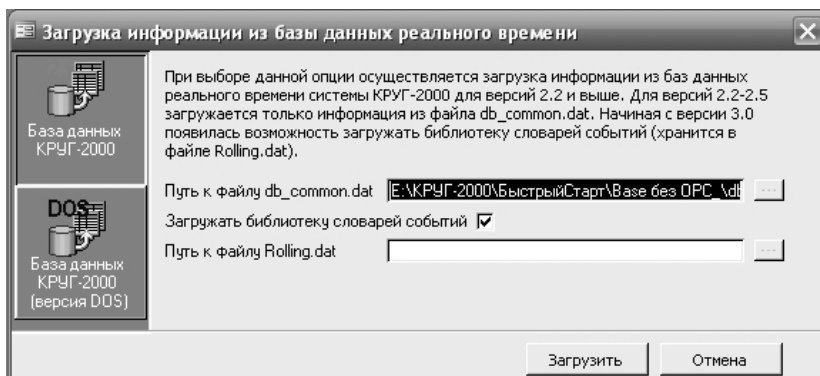


Рис. 3.3. Окно запроса пути загрузки БД

Главное меню Генератора БД представлен на рис. 3.4.

Главное меню Генератора БД системы Круг-2000 содержит следующие пункты.

Словари. Выбор данного пункта из главного меню осуществляется щелчком левой клавиши мыши на кнопке с названием Словари, при этом под выбранной кнопкой будет распахнуто подменю, содержащее все подчинённые опции меню в виде списка (иконок в левом меню) с надписями опций подменю.

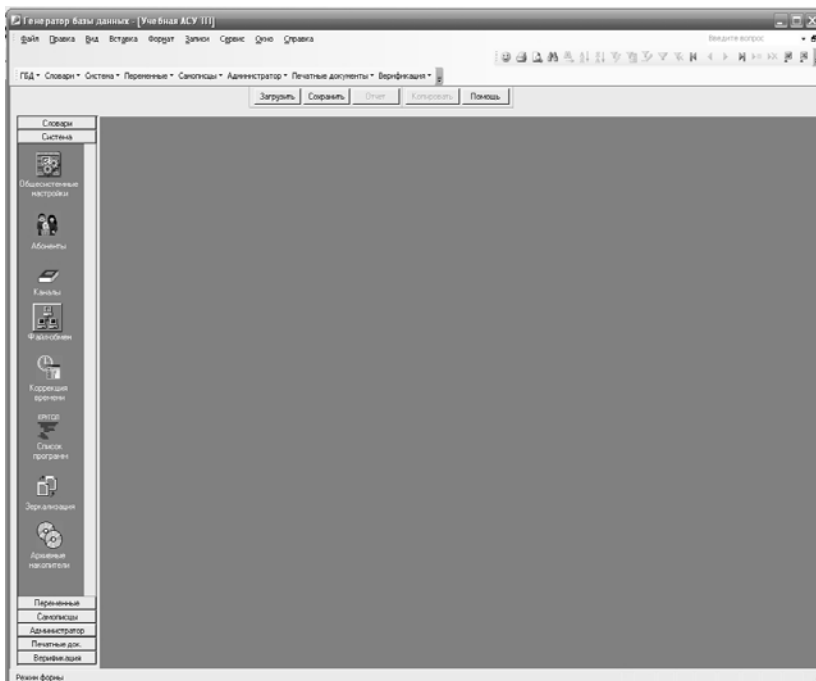


Рис. 3.4. Главное меню Генератора БД системы Круг-2000

Подменю СЛОВАРИ позволяет Вам:

- редактировать общесистемные словари:
 - словарь единиц измерений для аналоговых переменных и вещественных переменных ручного ввода;
 - словарь логических состояний для дискретных переменных и логических переменных ручного ввода;
 - словарь статусов;
 - словарь событий;
 - словарь наименований.
- создавать и редактировать таблицы нелинейности двух- и трёх-мерные. Эти таблицы понадобятся при генерации базы данных системы,
- настраивать цвета системы, используемые при отображении логических состояний входных и выходных дискретных переменных и цветов протокола сообщений системы.

Словарь единиц измерения предназначен для хранения списка используемых в системе единиц измерения, и применяется в дальнейшем, при описании аналоговых переменных базы данных и переменных ручного ввода вещественного типа. Максимальное количество единиц измерения – 65, максимальная длина единицы измерения – 10 символов.

Словарь логических состояний предназначен для хранения списка используемых в системе имен логических состояний дискретных переменных, и применяется в дальнейшем, при описании состояний для дискретных переменных базы данных и переменных ручного ввода логического типа. Максимальное количество логических состояний – 255, максимальная длина названия логического состояния – 10 символов.

Двухмерные таблицы нелинейности предназначены для описания зависимостей типа $Y = F(X)$ и могут использоваться в дальнейшем при обработке входного сигнала аналоговой входной переменной или в программах пользователя на языке КРУГОЛ. Максимальное количество таблиц – 32 767, точек в двухмерных таблицах – 9000.

Трёхмерные таблицы нелинейности могут использоваться в дальнейшем и непосредственно в УСО в программах пользователя, разработанных в среде КРУГОЛ. Для этого используются функции работы с таблицами нелинейности языков ST и FBD среды КРУГОЛ.

Подменю «Цвета системы» предназначено для настройки используемых по умолчанию цветов в системе реального времени. Данные цвета появляются в описателях динамических элементов, связанных с цветом отображения имен логических состояний для входных и выходных дискретных переменных и цветами отображения сообщений системы различных типов.

Словарь статусов и словарь наименований предназначены для описания различных событий, и их атрибутов (наименования, цвет текста, признак мигания, типа звуковой сигнализации и др.), и используются для задания типа и статуса события при создании словаря событий.

Система. Выбор пункта из главного меню осуществляется щелчком левой клавиши мыши на кнопке с названием Система, при этом под выбранной кнопкой будет распахнуто подменю, содержащее все подчинённые опции меню в виде списка (иконок в левом меню) с надписями опций подменю.

Подменю СИСТЕМА позволяет Вам:

- описать название системы;
- описать всех абонентов, входящих в структуру ПТК;
- описать параметры настройки адаптеров связи абонентов ПТК;

- описать характеристики каналов связи с абонентами ПТК;
- задать количество переменных всех типов, используемых в базе данных системы;
- описать параметры обмена данными с другими АСУ через механизм файл-обмена;
- задать коррекцию времени, которая состоит из следующих этапов:
 - описание групп корректоров времени;
 - описание свойств корректоров времени;
 - описание абонентов коррекции времени;
 - описание параметров переходов Зима-Лето;
- задать список программ на языке КРУГОЛ;
- задать параметры заркализации БД;
- задать архивные накопители.

При выборе иконки «Общесистемные настройки» на экран выводится форма для указания названия системы, настройки протокола событий, указания директории для архива печатных документов, заполнения количества переменных системы всех типов (рис. 3.5).

Таковыми переменными являются:

- входные аналоговые переменные (ВА);
- аналоговые выходные переменные (АВ);
- входные дискретные переменные (ВД);
- дискретные выходные переменные (ДВ);
- переменные ручного ввода вещественного типа (РВВ);
- переменные ручного ввода логического типа (РВЛ);
- переменные ручного ввода строкового типа (РВС).

Название системы

Учебная АСУ ТП

Количество переменных

Количество переменных типа ВА	10
Количество переменных типа АВ	0
Количество переменных типа ВД	3
Количество переменных типа ДВ	3
Количество переменных типа РВ(вещ.)	2
Количество переменных типа РВ(лог.)	0
Количество переменных типа РВ(стр.)	0
Общее количество переменных в БД	18

Настройка протокола событий

Максимальный размер оперативного протокола событий	10000
Кратность записи событий на диск	1
Квотировать событие при квотировании параметра	<input checked="" type="checkbox"/>
Квотировать параметр при квотировании события	<input checked="" type="checkbox"/>

Архив печатных документов

C:\Users

Рис. 3.5. Форма «Общесистемные настройки»

Программы, написанные пользователем для абонентов верхнего уровня системы (станция оператора, станция архивирования и другие), выполняются в реальном времени ядром Среды Разработки КРУГОЛ™. Предварительно программы пользователя должны быть откомпилированы в Среде Разработки КРУГОЛ™ и сохранены в директорию, в которой расположена БД системы. Откомпилированные программы пользователя сохраняются в виде файлов с расширениями «*.out» и «*.lab».

При выборе иконки «Список программ» на экран вызывается форма для формирования списка пользовательских программ на языке КРУГОЛ (рис. 3.6).

При описании списка программ необходимо заполнить следующие поля:

- номер – уникальный пользовательский код программы (целое положительное число);
- имя – уникальное пользовательское имя программы;
- приоритет – приоритет выполнения программы.

Переменные. Подменю ПЕРЕМЕННЫЕ позволяет описать все типы переменных, используемых в системе, к которым относятся:

- входные аналоговые переменные (ВА);
- аналоговые выходные переменные (АВ);
- входные дискретные переменные (ВД);
- дискретные выходные переменные (ДВ);
- переменная ручного ввода (РВ), состоит из следующих под-типов:
 - переменные ручного ввода вещественного формата (РВВ);
 - переменные ручного ввода логического формата (РВЛ);
 - переменные ручного ввода строкового формата (РВС).



Рис. 3.6. Форма описания списка программ

Входная аналоговая												
<div style="float: right; text-align: right;"> Загрузить Сохранить Отчет Копировать Помощь </div>												
Словари	№ переменной	№ канала	№ платы	№ входа	№ пере	№ канала	№ УСЧ в	Основной/Рез	Позиция	Имя 1	Имя 2	Длинная п
Система	1	0	0	0	0	0	0	0 Без резерв	U	Уровень жидкос	L жидкости	
Переменные	2	0	0	0	0	0	0	0 Без резерв	TI	Темпе ратура	T в резерв	
	3	0	0	0	0	0	0	0 Без резерв	PI	Давлен-трубоп	P в трубоп	
	4	0	0	0	0	0	0	0 Без резерв	TIC 01	Темпер	1 точки	
	5	0	0	0	0	0	0	0 Без резерв	TIC 02	Темпер	2 точки	
	6	0	0	0	0	0	0	0 Без резерв	PIC 01	Давлен	1 точки	
	7	0	0	0	0	0	0	0 Без резерв	PIC 02	Давлен	2 точки	
	8	0	0	0	0	0	0	0 Без резерв	FIC 01	Расход	1 точки	
	9	0	0	0	0	0	0	0 Без резерв	FIC 02	Расход	2 точки	
	10	0	0	0	0	0	0	0 Без резерв	DIC 01	Плотн	1 точки	
	* 11	0	0	0	0	0	0	0 Без резерв				

Рис. 3.7. Таблица описания входных аналоговых переменных

Задание атрибутов переменных осуществляется выбором пункта из Подменю ПЕРЕМЕННЫЕ щелчком левой клавиши мыши на кнопке с необходимым типом переменной, при этом открывается таблица со списком переменных данного типа и их атрибутами (рис. 3.7).

Для удобства работы с атрибутами переменной, список атрибутов можно отобразить в виде подчинённой формы. Для этого необходимо щелкнуть правой кнопкой мыши в области заголовка таблицы атрибутов и выбрать «Подчинённая форма/Форма» (рис. 3.8).

При работе с Генератором БД необходимо выполнять периодическое сжатие базы данных для оптимизации таблиц базы данных [22]. Сжатие базы данных осуществляется щелчком мыши на иконке ☺. При этом на экран будет выдан запрос о завершении работы, при щелчке на кнопке «Да» произойдёт закрытие базы данных и её сжатие. При щелчке на кнопке «Нет» – сжатия базы данных не произойдет. После выполнения сжатия на экране появится меню выбора файла базы данных для загрузки в генератор.

Самописцы. Описание самописцев выполняется после того, как описаны все типы переменных системы и конфигурация системы в целом.

Самописцы предназначены для непрерывной регистрации и хранения каких-либо значений параметров системы любых форматов в течение определённого интервала времени [22]. История параметров (оперативный тренд) по каждому самописцу хранится в поддиректории базы данных системы в виде отдельного файла с именем RibbonN.dat (где N – номер самописца). В оперативном тренде самописца хранится информация об истории изменения параметров за заданную в конфигурации самописца глубину хранения данных. Структурной единицей самописца является «перо самописца» – ссылка на какой-либо атрибут в таблицах переменных системы, значение которого требуется регистрировать с заданным периодом записи.

№ переменной	1	Начало шкалы	0
№ канала	0	Конец шкалы	100
№ платы	0	Гистерезис сигнализации	0,5
№ входа на плате	0	Время определения скорости изменения параметра	0
№ переменной в УСО	0	Граница сигнализации по скорости роста	0
№ УСО в канале	0	Граница сигнализации по скорости падения	0
№ канала в конвейере	0	Постоянная фильтра	0
Основной/Резерв	Без резерва	Максимально возможная (достоверная) скорость	0
Позиция	1	Тип замены недостоверного значения	0 - по шкале прибора
Имя1	Уровень	Рабочее значение переменной	0
Имя2	жидкости	Отсечка нуля	0
Длинное имя	Уровень жидкости в резервуаре	Снятие переменной с опроса	
Длинная позиция	жидкости	Снятие переменной с сигнализации	
Единица измерения	л	Снятие с опроса в СО	
Тип датчика	27 - м	Снятие с сигнализации в СО	
Тип датчика	0-5mA, 4-20mA, 0-10В, 0, 1	Текущее значение после преобразования (контроллер)	40
Тип датчика	Линейная	Текущее значение после (СО)	40
Начало шкалы тренда	0	№ алгоритма нестандарт обработки 1	1
Конец шкалы тренда	100	Вкл/выкл нестандарт обработки 1	<input checked="" type="checkbox"/>
Нижняя предаварийная граница	10	Тип регистра УСО	0
Нижняя предупредительная граница	20	Апертура	0
Верхняя предупредительная граница	80	№ алгоритма нестандарт обработки 4	0
Верхняя предаварийная граница	90		

Запись: 1 из 10

Рис. 3.8. Атрибуты входной аналоговой переменной в виде формы

При использовании сервера БД с функциями архивирования имеется возможность выполнять архивирование данных оперативных трендов в архивные тренды. Файлы архивных трендов сохраняются в директории, указываемой в свойствах архивных накопителей (смотри раздел «Архивы») с именами RibbonN_D_T.dat (где N – номер самописца, D – дата записи 1-й точки в архиве в формате ГГГГ_ММ_ДД, T – время записи 1-й точки в архиве в формате ЧЧ_ММ_СС).

В системе поддерживаются следующие типы самописцев:

- Базовые самописцы – цикл записи точек по переменным, входящим в самописец, определяется стандартными базовыми интервалами времени.

- Производные самописцы – цикл записи точек по переменным, входящим в самописец, определяется пользовательскими интервалами времени, описанными в форме «Интервалы времени». Производные самописцы формируются на основании данных соответствующих базовых самописцев. Данные производных самописцев не архивируются и хранятся в виде одной циклической ленты.

- Событийные самописцы – формируются только в моменты изменения значений технологических параметров. Событийные тренды позволяют в одном событийном самописце регистрировать изменения значений переменных, принадлежащих разным каналам, например, каналам «ОПС НДА-клиент» или «РС-контроллер».

Работа с подменю САМОПИСЦЫ состоит из следующих этапов:

- конфигурация базовых самописцев;
- описание перьев базовых самописцев;
- описание пользовательских интервалов времени;
- конфигурация производных самописцев;
- описание перьев производных самописцев;
- конфигурирование событийных самописцев;
- конфигурирование канальных самописцев;
- описание событийных самописцев.

Все переменные, которые необходимо регистрировать, рекомендуется группировать в самописцы в зависимости от требуемого периода записи точек и вне зависимости от привязки к участкам и группам системы.

При выборе иконки «Базовые самописцы», на экран вызывается табличная форма для работы со списком базовых самописцев, используемых в системе (рис. 3.9). Далее необходимо заполнить нужные поля таблицы базовых самописцев.

Базовые самописцы							
Загрузить Сохранить Отчет Копировать Помощь							
Словари	№ самописца	Имя самописца	Количество перьев	Количество точек	Кратность записи	Период записи	Единица записи
Система	1	Секундный самописец	14	1200	0	0	1 Новая секунда
Переменные	2	Минутный самописец	14	1200	0	0	1 Новая минута
Самописцы	3	Часовой самописец	14	1200	0	0	1 Новый час
	4						
	5						
	6		0	0	0	0	0

Рис. 3.9. Таблица конфигурации базовых самописцев

В верхней части формы отображается имя текущего самописца и основные параметры самописца (количество перьев, количество точек и период записи – являются информационными полями). Для перехода к другому самописцу необходимо выполнить щелчок мышью на кнопке разворачивания списка самописцев, расположенной справа от текущего имени таблицы, при этом раскроется список всех таблиц самописцев данного типа с полями «№», «Имя самописца». Выберите нужную строку из списка и щелкните на ней мышью, при этом будут выведены данные по выбранному самописцу. Другим вариантом является переход между самописцами с помощью навигатора записей в нижней строке формы.

При выборе иконки «Базовые перья», на экран выводится форма для описания перьев базовых самописцев (рис. 3.10).

Список всех перьев для текущего самописца выводится в виде таблицы. Перемещение между записями перьев осуществляется с помощью навигатора записей в нижней строке формы или с помощью клавиатуры.

Форма описания перьев самописцев предназначена для выполнения следующих операций:

- выбор формы заполнения перьев самописца (основная, переменная),
- изменение и добавление перьев в самописец,
- автоматический режим заполнения перьев самописца,
- удаление перьев самописца,
- копирование перьев самописца.

Автоматический режим заполнения перьев самописца предназначен для быстрого заполнения перьев самописца переменными выбранного типа и выполняется с помощью кнопки «Автозаполнение» (рис. 3.11).

Форма «Интервалы времени» предназначена для описания «пользовательских» интервалов времени, на основе которых пользователь может формировать самописцы с интервалами записи, отличными от стандартных интервалов времени, используемых в системе (например: сменные, суточные и др.). Самописцы, сформированные на основе описанных в данной форме интервалов времени, являются производными, в отличие от самописцев, использующих стандартные интервалы времени.

Базовые перья

Слезы Система Перья Сангопсыци

№ Имя сангопсыци
 Т Секундный сангопсыци
 Количество перьев 14
 Количество точек 1200
 Период записи 1 секунда

Автозаполнение Изменить Вид формы
 Изменить группу перьев Копирование Основная
 Удалить Переменная

Загрузить Сохранить Отчет Копировать Помощь

№ пера	Имя пера	Источник данных		Тип обработки		Источник данных по условию	
		Таблица	Запись	Атрибут	Атрибут	Таблица	Запись
1 сек. LI	Входная аналогов	1	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
2 сек. TI	Входная аналогов	2	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
3 сек. PI	Входная аналогов	3	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
4 сек. NAT Cost	Входная дискретне	1	Текущее значение в CO	Среднее за интервал	пусто	пусто	пусто
5 сек. ZD1 0	Входная дискретне	2	Текущее значение в CO	Среднее за интервал	пусто	пусто	пусто
6 сек. ZD1 3	Входная дискретне	3	Текущее значение в CO	Среднее за интервал	пусто	пусто	пусто
7 сек. Time Har	RV Составная	1	Текущее значение	Среднее за интервал	пусто	пусто	пусто
8 сек. TIC 01	Входная аналогов	4	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
9 сек. TIC 02	Входная аналогов	5	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
10 сек. PIC 01	Входная аналогов	6	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
11 сек. PIC 02	Входная аналогов	7	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
12 сек. FIC 01	Входная аналогов	8	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
13 сек. FIC 02	Входная аналогов	9	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто
14 сек. DIC 01	Входная аналогов	10	Текущее значение посл	Среднее за интервал	пусто	пусто	пусто

Запись: 1 из 14
 Запись: 1 из 3

Администратор
 Печать док.
 Верификация

Рис. 3.10. Форма описания перьев базовых сангопсыци

Рис. 3.11. Форма описания параметров пера «Автозаполнение»

Конфигурация производных самописцев и описание перьев производных самописцев осуществляется аналогично базовым.

Администратор. Подменю АДМИНИСТРАТОР позволяет описать следующие параметры системы:

- функции доступа;
- группы доступа;
- права доступа к таблицам настройки переменных;
- формирование списка атрибутов для таблиц настройки переменных.

В системе можно организовать до 64 функций доступа к БД и системе в целом, которые могут быть назначены определённым группам пользователей системы. Непосредственная привязка функций доступа к реакциям системы осуществляется на этапе создания графического проекта системы.

При выборе пункта меню «Функции доступа» (рис. 3.12) на экран вызывается форма для работы с функциями доступа, которая позволяет отредактировать название выбранной функции доступа. Удаление или добавление функций не возможно.

При выборе пункта меню «Группы доступа» на экран вызывается форма для работы с группами доступа к базе данных и функциям системы (рис. 3.13). Группы доступа предназначены для организации в системе групп пользователей, имеющих одинаковые права доступа. Количество групп пользователей не ограничено. Пользователи могут входить только в одну группу пользователей. Добавление пользователей в существующие группы осуществляется с помощью программы «Менеджер пользователей».

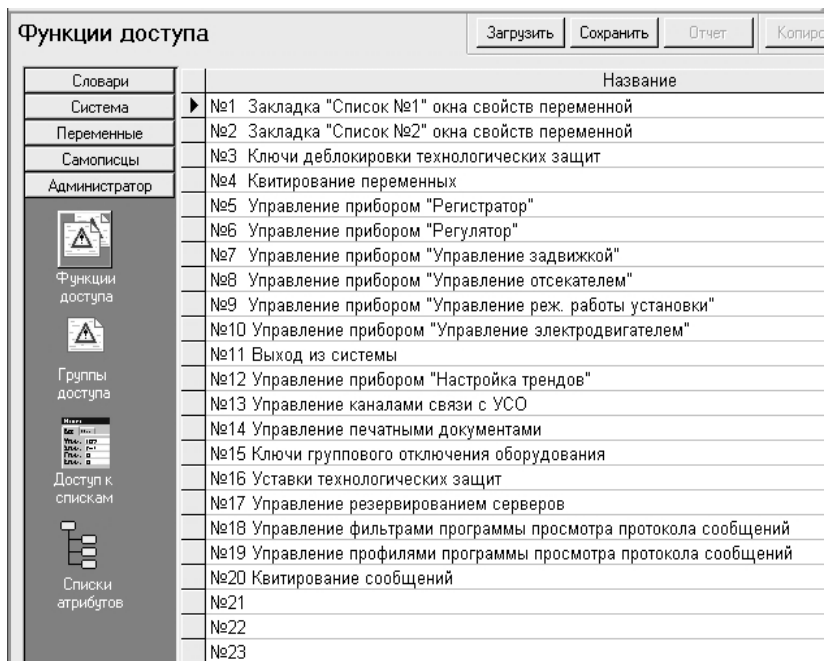


Рис. 3.12. Форма работы с функциями доступа

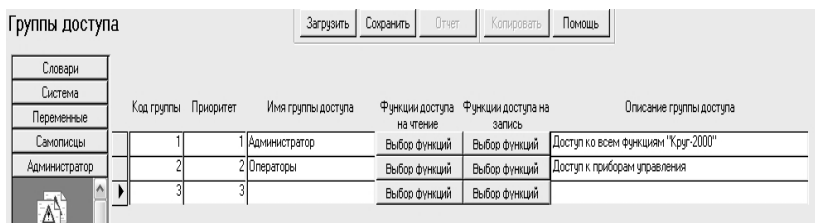


Рис. 3.13. Форма описания групп доступа



Рис. 3.14. Форма описания прав доступа для групп пользователей

В форме описания групп доступа для каждой записи в полях «Функции доступа на чтение» и «Функции доступа на запись» расположены кнопки «Выбор функций», с помощью которых вызываются формы для установки права доступа на чтение (запись) для той или иной группы пользователей (рис. 3.14).

В графическом интерфейсе станций оператора/архивирования возможен вызов таблиц настройки переменных для просмотра и изменения свойств переменных. При выборе иконки «Права для списков» на экран вызывается форма назначения прав доступа на чтение и запись к соответствующим закладкам таблицы настройки для переменных всех типов (рис. 3.15).

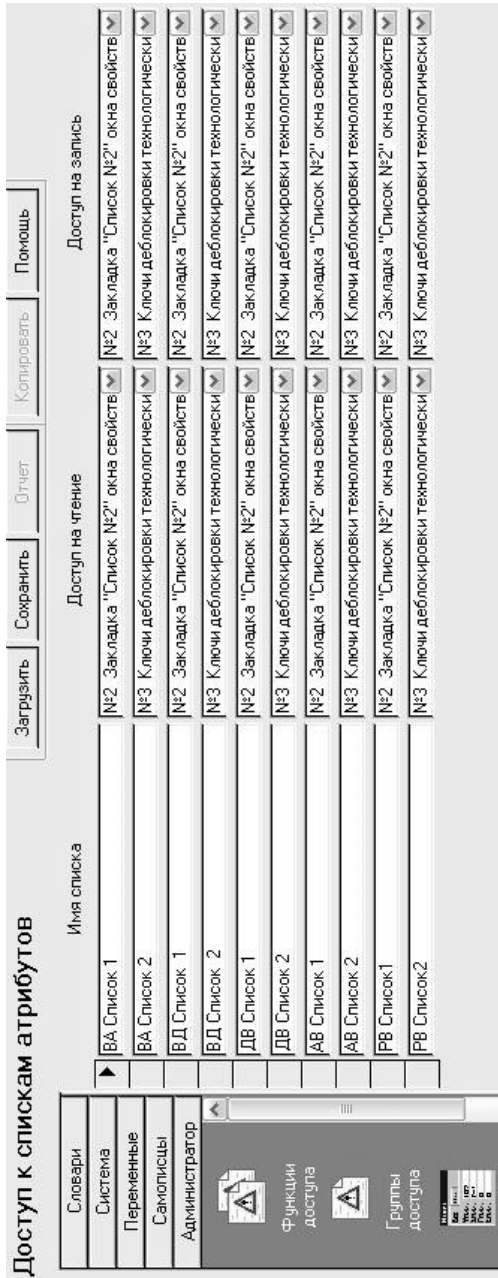


Рис. 3.15. Форма для работы с правами доступа к закладкам таблицы настройки переменных

Форма «Права доступа для списка свойств» представлена в виде таблицы с полями:

- Имя списка – имена закладок таблицы настройки соответствующих типов переменных по умолчанию.

- Права на чтение – функция доступа на чтение к данной закладке, выбирается из выпадающего списка функций доступа на чтение, сформированного предварительно. При наличии доступа к данной функции в системе реального времени пользователь может просмотреть список атрибутов и их значений для соответствующей закладки (списка) таблицы настройки переменных без возможности изменения (запрет изменения атрибута индицируется символом * в начале строки с именем соответствующего атрибута).

- Права на запись – функция доступа на запись к данной закладке, выбирается из выпадающего списка функций доступа на запись, сформированного предварительно. При наличии доступа к данной функции в системе реального времени пользователь может просмотреть и изменить в реальном времени значения атрибутов в соответствующей закладке (списке) таблицы настройки переменных (возможность изменения предоставляется только для атрибутов, не помеченных символом *).

Печатные документы. В системе КРУГ-2000 могут быть сформированы и выведены на печать следующие типы печатных документов:

- Протокол событий – формируется и печатается автоматически согласно параметрам описания в генераторе базы данных, также имеется возможность выборочной печати пользователем.

- Режимный лист – формируется и выводится на печать с помощью технологического языка программирования КРУГОЛ на основе предварительно созданных видеок кадров печатных документов в графическом проекте системы.

- Протокол пред- и послеаварийных ситуаций – формируется и выводится на печать с помощью технологического языка программирования КРУГОЛ на основе предварительно созданных видеок кадров печатных документов в графическом проекте системы.

- Отчёт произвольной формы – формируется и выводится на печать с помощью технологического языка программирования КРУГОЛ на основе предварительно созданных видеок кадров печатных документов в графическом проекте системы.

- Копия видеок кадра – формируется и выводится на печать вручную оператором или с помощью технологического языка программирования КРУГОЛ на основе предварительно созданных видеок кадров в графическом проекте системы.

Конфигурация печатных документов выполняется после того, как описаны все типы переменных системы, выполнена конфигурация перьев самописцев, которые будут использованы в печатных документах, а также созданы с помощью генератора динамики печатные документы в виде схем типа «Печатный документ». Данное меню предназначено для описания свойств печатных документов, настроек принтеров и параметров страниц.

Работа с подменю ПЕЧАТНЫЕ ДОКУМЕНТЫ состоит из следующих этапов:

- конфигурация параметров страниц;
- конфигурация настроек печати;
- конфигурация печатных документов;
- конфигурация описателей протокола событий.

В форме «Настройки печати» (рис. 3.16) необходимо заполнить следующие поля:

- номер – порядковый номер принтера в системе (каждый принтер должен иметь индивидуальный номер);
- имя – пользовательское имя принтера;
- системное имя – системное имя принтера, вводится вручную или выбирается из выпадающего списка предварительно установленных на компьютере принтеров;
- время обработки ошибки – время обработки ошибки от принтера в секундах (при отсутствии бумаги в принтере и т.п.). При превышении указанного времени в протокол событий выдается соответствующее сообщение и печатные документы перенаправляются на другой принтер, указанный в качестве резервного.

В форме «Параметры страницы» задаются форматы страниц, используемых в системе при печати (рис. 3.17). В форме «Параметры страницы» задаются форматы страниц, используемых в системе при печати (рис. 3.17).

При выборе иконки «Печатные документы» на экран вызывается окно диалога для выбора файла графического проекта с расширением «.kgr», который будет использоваться при работе системы и в котором хранятся мнемосхемы печатных документов. После выбора файла графического проекта и нажатии кнопки «Открыть» Генератор БД считывает информацию о мнемосхемах проекта типа «Печатный документ», и на экране отображается форма для описания параметров данных печатных документов. При отказе от выбора графического проекта – нажатии кнопки «Отмена» – используются данные о печатных документах, хранящиеся в генераторе базы данных от предыдущей загрузки графического проекта системы.

Принтеры				
Номер	Имя	Системное имя	Время обработки ошибки, сек	
▶	1	Принтер	HP LaserJet 1200 Series PCL 6	0
*▶	0			0

Запись: [←] [↩] [] 1 [▶] [▶▶] [▶▶▶] из 1

Путь к файлам для клиентов печати

Станция оператора: C:\Temp

Печать роллинга по выбору: C:\Temp

Автоматическая печать протокола сообщений: C:\Temp

Сервер печати

Кол-во принтеров в системе: [] 1

Автоматическая печать буфера после сбоя:

Максимальное кол-во документов в очереди: [] 10

Рис. 3.16. Форма описания настроек печати

Вид страницы :

A4 (книга)

Имя вида страницы : A4 (книга)

Номер вида страницы : [] 1

Поля :

Верхнее, мм : [] 20

Нижнее, мм : [] 20

Левое, мм : [] 20

Правое, мм : [] 20

По умолчанию ...

Размер бумаги :

A4 210 x 297 mm

Ориентация :

Книжная

Альбомная

Запись: [←] [↩] [] 1 [▶] [▶▶] [▶▶▶] из 2

Рис. 3.17. Форма описания параметров страницы

В системе возможно создание различных описателей протоколов событий, которые позволяют формировать и выводить на печать сообщения системы с учётом различных заданных фильтров и направлять их на разные принтеры (например: протоколы аварийных сообщений, общие протоколы сообщений и т.п.).

Верификация системы. Верификация системы выполняется после завершения генерации системы в целом (или после завершения

отдельных шагов генерации системы) и предназначена для получения сводной информации по системе, обнаруженных ошибках при генерации системы и статистики по отдельным функциям системы.

Работа с подменю ВЕРИФИКАТОР состоит из следующих этапов:

- конфигурация верификатора;
- запуск верификатора;
- статистика по переменным;
- статистика по каналам;
- статистика по нелинейности;
- статистика по самописцам.

При выборе иконки «Конфигурация верификатора» на экран вызывается форма для описания параметров верификации системы (рис. 3.18), которая состоит из следующих закладок:

- проверка БД на дублирование адресов переменных;
- проверка БД на дублирование позиций переменных;
- проверка БД при вводе данных в ячейку.

The image displays three screenshots of a software configuration window for system verification. Each window has three tabs: 'Адреса', 'Позиции', and 'Ввод данных'.

Top-left screenshot (Address tab):

- Разрешить проверку на дублирование адресов
- Не проверять переменные с № канала = 0
- Тип проверки:
 - № канала, № платы, № входа
 - № канала, № в УСО

Top-right screenshot (Data Input tab):

- Разрешить проверку на дублирование позиций
- Проверка:
 - по всем типам технологических переменных
 - внутри одного типа технологических переменных
 - не проверять переменные с пустыми позициями
- Проверить только:
 - входные аналоговые переменные (ВА)
 - выходные аналоговые переменные (АВ)
 - входные дискретные переменные (ВД)
 - выходные дискретные переменные (ДВ)
 - переменные ручного ввода вещественного типа (РВВ)
 - переменные ручного ввода логического типа (РВЛ)
 - переменные ручного ввода типа строка (РВС)

Bottom screenshot (Data Input tab):

- Проверять данные при вводе в ячейку
- Проверять измененные записи при закрытии формы
- Полностью проверять таблицы при закрытии формы
- Не проверять ввод данных

Рис. 3.18. Формы описания параметров верификации системы

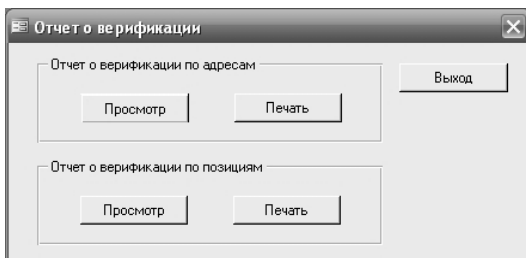


Рис. 3.19. Форма работы с отчётами верификации базы

При выборе иконки «Запуск верификатора» выполняется верификация БД по переменным в соответствии с заданной конфигурацией верификатора. После окончания процесса верификации на экран выводится форма работы с отчётами верификации БД (рис. 3.19), которая предназначена для следующих режимов работы:

- просмотр и печать отчётов верификации по адресам переменных;
- просмотр и печать отчётов верификации по позициям переменных.

При выборе иконки «Статистика» на экран вызывается форма с закладками статистики: по каналам (для каждого типа переменных) (рис. 3.20), для каждого типа нелинейности, по самописцам.

Статистика по каналам Статистика по нелинейности Статистика по самописцам

	Все типы переменных				Входная аналоговая				Аналоговая выходная				Входная дискретная				Дискретная выходная			
	всего	физич.	вирт.	резер.	всего	физич.	вирт.	резер.	всего	физич.	вирт.	резер.	всего	физич.	вирт.	резер.	всего	физич.	вирт.	резер.
Всего в БД	18	0	18	0	10	0	10	0	0	0	0	0	3	0	3	0	3	0	3	0
Канал № 0	18	0	18	0	10	0	10	0	0	0	0	0	3	0	3	0	3	0	3	0
Канал № 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Просмотр Печать

Рис. 3.20. Форма работы со статистикой

Таким образом, создание БД системы реального времени системы Круг-2000 состоит из последовательности шагов, выполняемых при работе с генератором.

Этап 1. Генерация словарей системы.

На данном этапе выполняются действия по созданию и редактированию словарей системы, данные из которых используются при генерации БД по типам переменных. Кроме словарей системы, на данном этапе выполняется генерация таблиц нелинейности, необходимых для работы системы, а также настройка цветов системы, используемых при отображении логических состояний входных и выходных дискретных переменных и цветов протокола сообщений системы.

Этап 2. Конфигурация системы.

На данном этапе выполняются действия по описанию конфигурации системы, к которым относятся: название системы, описание параметров абонентов ПТК, описание параметров адаптеров связи абонентов ПТК, описание параметров каналов связи с УСО, описание перечня программ технологического языка КРУГОЛ и др.

Этап 3. Описание переменных базы данных.

На данном этапе выполняются действия по описанию переменных базы данных всех типов.

Этап 4. Генерация самописцев.

На данном этапе выполняются действия по конфигурации самописцев и описанию переменных, входящих в каждый из самописцев.

Этап 5. Администрирование системы.

На данном этапе выполняются действия по администрированию системы, к которым относятся: конфигурация групп пользователей, конфигурация функций доступа к системе реального времени, конфигурация прав доступа к спискам свойств переменных и др.

Этап 6. Конфигурация печатных документов.

На данном этапе выполняются действия по конфигурации функций абонентов печати и печатных документов, к которым относятся: описание принтеров и параметров печати, описание параметров страниц, используемых при формировании печатных документов, описание печатных документов, формируемых системой реального времени, описание параметров протокола событий системы реального времени.

Этап 7. Верификация системы.

На данном этапе выполняются действия по верификации созданной базы данных системы и документированию результатов верификации (выполняется при необходимости).

Этап 8. Создание БД системы реального времени.

На данном этапе выполняются действия по сохранению БД в виде файлов, используемых в системе реального времени серверами БД.

3.3. ГЕНЕРАТОР ДИНАМИКИ СИСТЕМЫ КРУГ-2000

Программа Генератор динамики (ГД) служит для формирования графического интерфейса Станции Оператора.

Алгоритм работы с ГД приведён на рис. 3.21.

Если Графическая База Данных ещё не создана, то работу в ГД начните с создания проекта.

Если проект уже существует и описан в поле *Имя файла графической базы данных* Менеджера задач SCADA КРУГ-2000 (рис. 3.22), то при вызове ГД проект загружается вместе с программой и работу в ГД следует начинать с загрузки мнемосхемы для редактирования или создания новой мнемосхемы.

Перед вызовом программы Станция Оператора в проекте обязательно надо создать хотя бы один рабочий стол и в нём переход на мнемосхему.



Рис. 3.21. Алгоритм работы с ГД

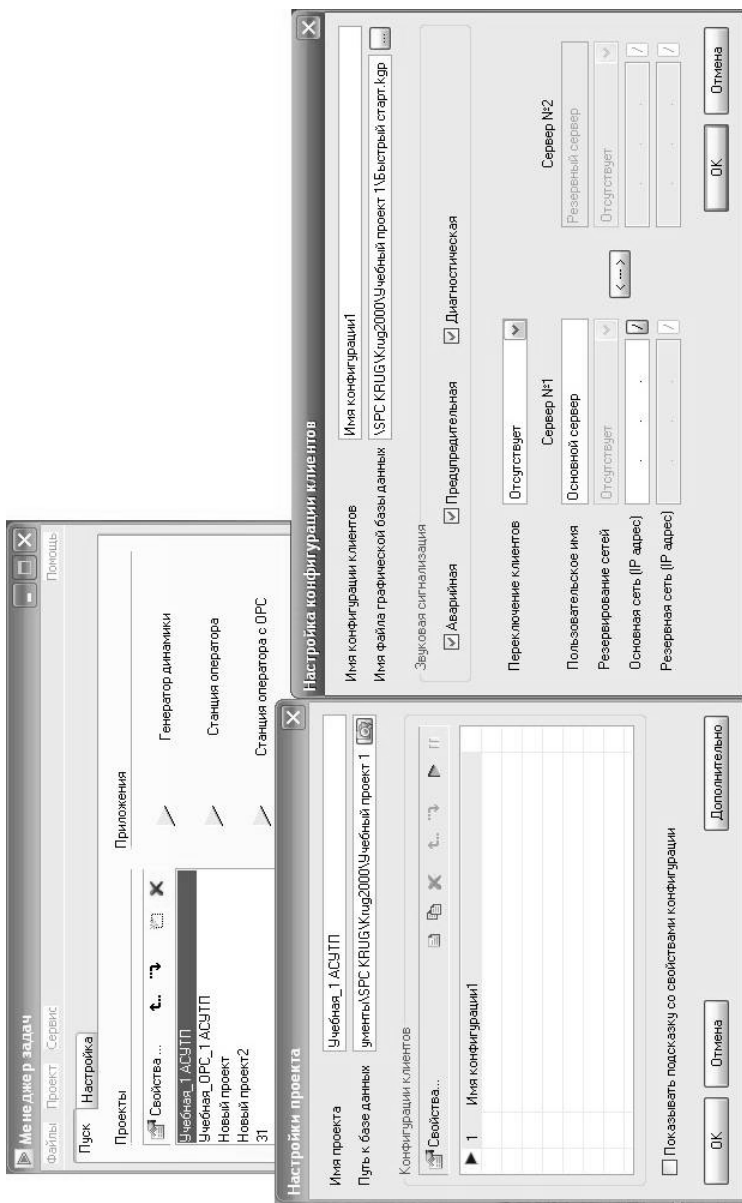


Рис. 3.22. Настройка проекта в Менеджере задач SCADA KРУГ-2000

Интерфейс пользователя ГД. На рисунке 3.23 показан общий вид рабочего стола ГД. Выноски на рисунке соответствуют следующим элементам рабочего стола.

1. Строка заголовка. Содержит название исполняемой программы и имя активной мнемосхемы.

2. Строка системного меню. Выбор пункта меню выводит на экран список команд, которые служат для управления соответствующим окном.

3. Панель инструментов. Содержит набор кнопок, дублирующих команды системного меню.

4. Кнопка *Свернуть/Распахнуть рабочее окно*. Сворачивает рабочее окно, оставляя только кнопку в панели Задачи Windows для свернутой программы или минимизированные окна мнемосхем в *Рабочем окне* ГД для свернутых мнемосхем.

5. Кнопка *Изменить размер окна*. Изменяет размер окна, уменьшая его, или разворачивает его во весь экран. После нажатия на кнопку она изменяет свой вид.

6. Кнопка закрытия окна. Служит для закрытия окна мнемосхемы или программы. При закрытии программы будет выведен запрос на сохранение проекта.

7. Рабочее окно – область кадра. Предназначено для расположения объектов.

8. Полосы прокрутки. Горизонтальная и вертикальная полосы прокрутки служат для передвижения рабочей области по горизонтали и вертикали.

Чтобы в строке заголовка назначить или отменить видимость и работу кнопок 4, 5, 6 (на рис. 3.23) необходимо настроить проект. Для этого необходимо открыть *Окно проекта* и нажать левой клавишей мыши на первой строке окна, где указан путь к проекту. В *Окне свойств* появятся свойства, относящиеся ко всему проекту (рис. 3.24).

Закладка *Главное*. Строка *Присутствие заголовка* отвечает за заголовок окна *Станции оператора*. *Да* – заголовок есть, *Нет* – заголовок отсутствует.

Строка *Присутствие системного меню* отвечает за кнопку закрытия окна. *Да* – кнопка есть, *Нет* – кнопка отсутствует. Соответственно, при отсутствии кнопки невозможно закрыть Станцию оператора с помощью пункта *Выход* меню *Файл* или с помощью кнопок Alt+F4. Аналогичным образом можно настроить работу остальных кнопок и элементов меню.

В закладке *Звук* описывается звуковая сигнализация проекта, назначаемая при запуске *Станции оператора*.

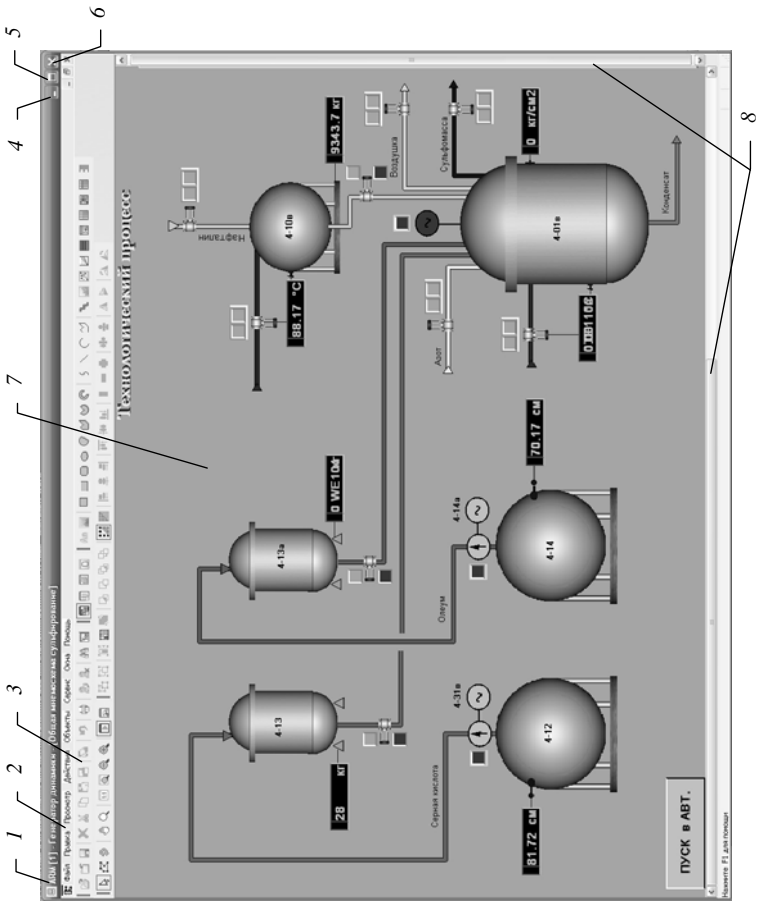


Рис. 3.23. Общий вид ГД

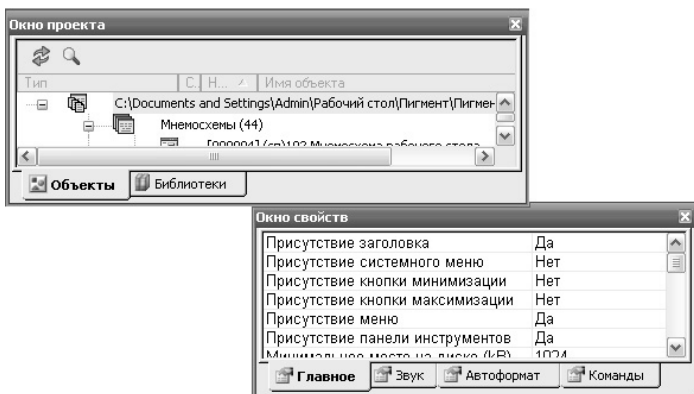


Рис. 3.24. Окно свойств проекта

В закладке *Команды* можно назначить клавиши переключения между окнами и между мнемосхемами.

Для определения реакций, относящихся ко всему проекту (квотирувание, вызов мнемосхемы в поле рабочего стола, вызов прибора и т.д.) необходимо в *Окне проекта* выделить строку *Мнемосхемы* и открыть *Окно свойств* (рис. 3.25). События, описанные в данных реакциях, имеют наивысший приоритет по отношению к событиям, назначенным на конкретную мнемосхему, а тем более динамическим элементам любой мнемосхемы.

Вторыми по приоритету идут события, назначенные на определённую мнемосхему. И последние по приоритету идут события, относящиеся к элементам на мнемосхеме.

Список доступных событий при назначении функций реакции на весь проект ограничен всего двумя событиями: нажатие клавиши (одной клавиши) и отжатие клавиши (одной клавиши).

Доступ к реакциям, относящимся к конкретной мнемосхеме, осуществляется нажатием левой клавиши мыши в *Окне проекта* на имени любой мнемосхемы. В *Окне свойств* появляются три закладки со свойствами мнемосхемы, кроме мнемосхем типа *Шаблон*, у которых закладка *Реакции* отсутствует (рис. 3.26).

В закладке *Реакции* можно описать функции реакции, относящиеся только к выбранной мнемосхеме, например, переход на смежную по структурной схеме мнемосхему стандартными клавишами (влево/вправо, вверх/вниз, листание вперед/назад, Home/End) вызовом приборов данной мнемосхемы, включение/отключение кнопок данной мнемосхемы.

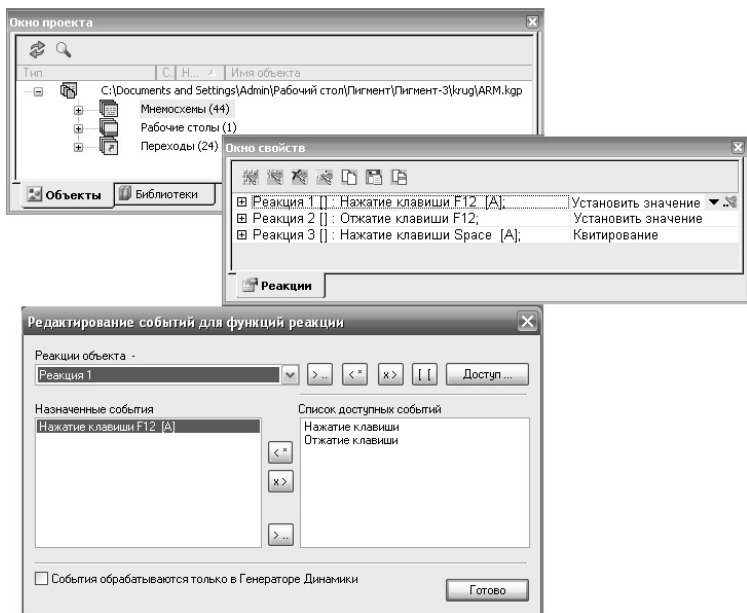


Рис. 3.25. Окно свойств мнемосхем проекта

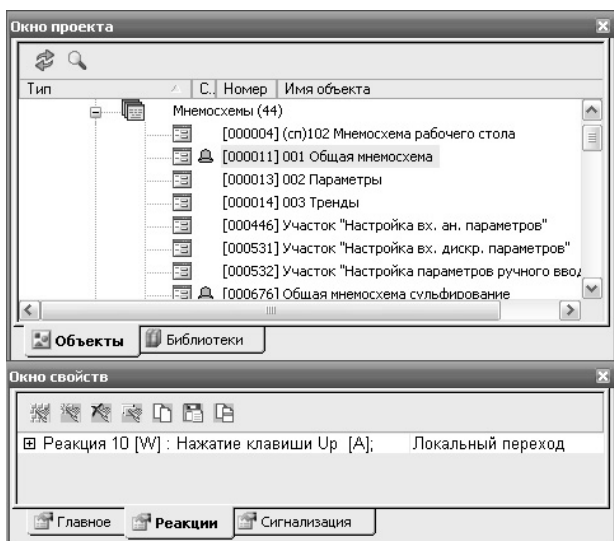


Рис. 3.26. Окно свойств конкретной мнемосхемы

В закладке *Главное* описаны свойства выбранной мнемосхемы.

Если мнемосхеме назначена сигнализация, то третья закладка называется *Сигнализация*, если нет – *Переменные*. Данный атрибут мнемосхемы формируется как обобщённый признак сигнализации по всем переменным, включённым на сигнализацию на данной мнемосхеме.


















В данной закладке отображается список переменных, находящиеся на конкретной мнемосхеме, в скобках указано количество ссылок переменной, используемых в элементах динамики на данной мнемосхеме.









Цвет колокольчика в строке переменной показывает, поставлена ли на сигнализацию переменная на данной мнемосхеме. Нажатием левой клавиши мыши на колокольчик можно изменить включение сигнализации: серый – не поставлена, другие цвета – поставлена.

Типы объектов графической базы данных (ГБД).

1. *Графический элемент* (примитив) (ГЭ). Элементарный графический объект, функционирующий на основе определённого набора собственных атрибутов и не связанный с переменными Общей Базы Данных, создаваемыми ГБД.

Кнопки для создания графических элементов в ГД:

-  – прямоугольник.
-  – объёмный прямоугольник.
-  – скругленный прямоугольник.
-  – эллипс.
-  – сегмент.
-  – многоугольник.
-  – сектор.
-  – большая дуга.
-  – кривая.
-  – линия.
-  – дуга.
-  – ломаная.
-  – труба.
-  – тренд.
-  – событийный тренд.
-  – режимный тренд.
-  – протокол событий.

-  – SQL – таблица.
-  – тренд в табличном виде.
-  – событийный тренд в табличном виде.
-  – расширенный тренд в табличном виде.
-  – шкала.
-  – текст.
-  – изображение.
-  – динамический шаблон.

2. *Составной элемент (СЭ)* – сложный графический объект, состоящий из группы графических элементов (ГЭ), объединённых в единое целое с помощью пункта *Группировать* подменю *Действия* Главного меню. Составной элемент имеет обобщённые свойства, в которые входят свойства отдельных графических элементов.

3. *Функция преобразования* – это зависимость свойства графического элемента или составного элемента от комбинации атрибутов переменных базы данных по определённому правилу, возникает при назначении элементу динамики. Для различных свойств элемента списки возможных для назначения динамик различны и могут включать следующие типы динамик: числовое, числовое из интервала, значение в текст, логическое, комбинация логических, преобразование координат и присвоить значение.

4. *Функция реакции* – это функция обработки действий пользователя. Под действием пользователя понимается нажатие или отжатие клавиши на клавиатуре, нажатие или отжатие правой или левой клавиши мыши, попадание или выход курсора мыши за границы элемента, последовательный набор (нажатие подряд до четырёх клавиш клавиатуры).

5. *Динамический элемент (ДЭ)* – это графический или составной элемент, состояние которого изменяется во времени. Изменение состояния ДЭ связано с изменением значений переменных БД или с изменением состояния других графических объектов. Для ДЭ можно назначить функции реакции на действия пользователя.

6. *Динамический прибор (ДПР)* – составной динамический элемент. ДПР состоит из комбинации ДЭ, ГЭ и СЭ. При объединении выделенных элементов в прибор все ссылки группируются по переменным, которым они назначены, и при копировании *Динамического прибора* назначаются только переменные, к которым будет привязан данный прибор, а все атрибуты внутри ссылок сохраняются, как в сформированном приборе.

7. *Мнемосхема* является элементом ГБД. Существует три типа мнемосхем: мнемосхема, печатный документ, шаблон прибора.

Мнемосхема – составной динамический прибор, состоящий из комбинации ДПР, ДЭ, ГЭ и СЭ, и имеющий собственное окно. Мнемосхема является основой, при изменении её типа (*Печатный документ* и *Шаблон прибора*) ей добавляются дополнительные свойства.

Мнемосхеме может быть назначена сигнализация. При назначенной сигнализации можно в списке переменных, для которых существуют динамики на данной мнемосхеме, выбирать те переменные, при срабатывании сигнализации по которым будет формироваться обобщённый признак сигнализации мнемосхемы.

Печатный документ – мнемосхема, сформированная специально для создания печатного документа. В системе реального времени печатный документ формируется с помощью программы КРУГОЛа по какому-либо условию и направляется в архив. Далее его можно распечатать.

Шаблон прибора – это мнемосхема, для которой меняется привязка в реальном времени (при вызове прибора). Привязка – это набор входных/выходных параметров прибора. Входной/выходной параметр – это совокупность ссылок на переменную БД внутри прибора, заменяемых на адрес одной переменной внутри прибора. При вызове прибора адреса переменных изменяются, и прибор настраивается на указанные для него переменные.

При вызове прибора все таблицы мнемосхемы-шаблона копируются, после чего происходит замена всех скопированных ссылок на текущие ссылки вызова прибора. Далее прибор ведёт себя как обычная мнемосхема, но при этом зависит от мнемосхемы родителя:

- если родитель закрывается, то закрываются и все приборы;
- мнемосхема прибора не может закрыться мнемосхемой родителя, т.е. прибор должен быть расположен поверх родителя.

После закрытия мнемосхемы все таблицы и ссылки удаляются.

8. *Рабочий стол* – совокупность мнемосхем, при помощи которых организуется внешний вид главного окна программы. Расположение элементов рабочего стола (мнемосхем) настраивается пользователем.

9. *Переход* – это объект ГБД, описывающий состояние и положение мнемосхемы внутри рабочего стола при её открытии, а также имеющиеся существующих рабочих столов, на которые возможны переходы.

Для выбора пункта меню ГД необходимо подвести курсор к выбираемому пункту меню, при этом выбираемый пункт будет выделен. В таком положении курсора необходимо нажать левую клавишу мыши. При этом появляется выпадающее подменю. Выбор пункта подменю

осуществляется аналогично вышеописанному. Для отмены выбранного подменю необходимо вывести курсор за пределы подменю и нажать любую клавишу мыши.

Для выбора необходимого инструмента нужно подвести курсор к кнопке инструмента, при этом вокруг выбираемой кнопки появится контур, а через небольшой промежуток времени рядом с кнопкой появится подсказка о выполняемом данным инструментом действии. В таком положении курсора необходимо нажать левую клавишу мыши. Если рисунок кнопки не раскрашен, то данный инструмент не доступен для данного объекта или в данной версии ГД.

Выбор элемента схемы. Под элементом схемы понимается ГЭ, СЭ, ДЭ и ДПР. Для выбора элемента схемы необходимо подвести курсор к выбираемому элементу и нажать левую клавишу мыши. Вокруг выбранного элемента появятся модификаторы размера (квадратики, расположенные по контуру). Нажав на один из модификаторов, можно изменить размер элемента, при этом курсор будет иметь вид крестика. Если нажимать на угловые модификаторы, то размер элемента изменяется и по горизонтали, и по вертикали, если нажимать на модификаторы, расположенные в серединах сторон контура, то размер элемента изменяется только в одном направлении.

Выделенный элемент можно перемещать в рабочем окне, когда внутри выделения модификаторами курсор принимает форму стрелки, необходимо нажать левую кнопку мыши и, удерживая ее, мышью переместить элемент на новое место.

Выделенный элемент можно перемещать с помощью клавиш управления курсором. Изменение размера выделенного элемента происходит при нажатой клавише **Shift** с помощью клавиш управления курсором в горизонтальном и вертикальном направлении.


Выбрать элемент можно, выделив область экрана курсором мыши при нажатой левой клавише мыши. Прямоугольник при этом должен быть больше выбираемого объекта. При правильной обводке вокруг выбираемого элемента появятся модификаторы.


Для копирования элемента необходимо установить курсор на выделенный элемент, нажать клавишу **Ctrl** и при нажатой левой клавише мыши произвести перемещение элемента до нужного места. Все свойства элемента будут также скопированы.

Выбрать группу элементов можно, обведя область с элементами прямоугольником при нажатой левой клавише мыши. Если выбирается группа элементов, то вокруг каждого из элементов появятся модификаторы белого цвета, а один из элементов будет выделен модификаторами чёрного цвета – этот элемент является выбранным по отношению

ко всем выделенным. Для того, чтобы добавить элемент к выбранной группе, необходимо выбрать элемент одним из вышеописанных методом с нажатой клавишей **Shift**. Нажатие левой клавиши мыши с нажатой клавишей **Shift** на модификаторах одного из выбранных элементов приведёт к отмене выделения данного элемента.

Работа с файлами. При входе в системное меню пункт *Файл* можно: создать проект или открыть его, если он уже был создан, проверить данные ГБД на непротиворечивость, распечатать мнемосхему (рис. 3.27). На экране появится подменю, из которого необходимо выбрать нужный пункт.

Для открытия проекта можно воспользоваться кнопкой . При выборе пункта *Открыть проект* на экране появится окно открытия проекта. В данном окне проекты обозначаются файлами с расширением «.kgr». Для открытия проекта необходимо выбрать имя проекта и нажать клавишу *Открыть* или дважды щелкнуть левой клавишей мыши на имени открываемого проекта. Для выхода из данного окна без открытия проекта нажмите клавишу *Отмена*.

Для создания нового проекта можно воспользоваться кнопкой . При выборе данного пункта на экране появится окно создания проекта. В строке ввода имени файла проекта необходимо задать имя проекта и затем нажать клавишу *Создать* (рис. 3.28).

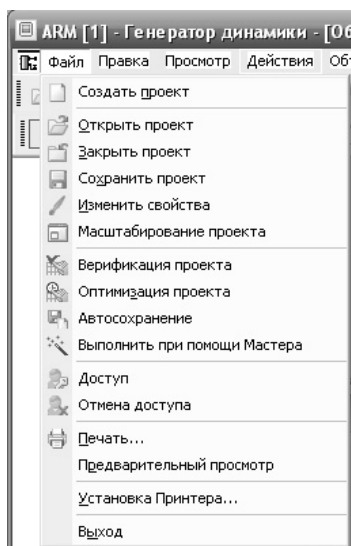


Рис. 3.27. Меню Файл



Рис. 3.28. Создание нового проекта

После создания или открытия проекта на экране появится *Окно проекта*. Из данного окна можно загружать объекты проекта, такие как мнемосхемы, рабочие столы или переходы, а также библиотеки. Для загрузки объектов необходимо выбрать нужный тип объекта и щелкнуть левой клавишей мыши, при этом развернётся список объектов выбранного типа. Двойной щелчок мыши на выбранном из списка объекте приводит к его загрузке.


Если открыто *Окно свойств*, то одинарный щелчок на имени объекта приведёт к заполнению *Окна свойств* свойствами выбранного объекта.



Верификация проекта – проверка на непротиворечивость созданной ГБД.

Для проверки необходимо выбрать из системного меню пункт меню *Файл*, а в нём пункт *Верификация проекта*.

Оптимизация проекта – упорядочивание существующих элементов, удаление связей с уже несуществующими элементами созданной ГБД. Всё это приводит к уменьшению размера файла проекта и ускоряет работу Станции оператора.

Для проверки необходимо выбрать из системного меню пункт меню *Файл*, а в нём пункт *Оптимизация проекта*.

Выбранную и загруженную мнемосхему можно распечатать с помощью принтера. Перед печатью мнемосхемы рекомендуется включить режим имитации для того, чтобы поля заполнились динамикой .

Для предварительного просмотра макет листа с мнемосхемой необходимо в меню *Файл* выбрать пункт меню *Предварительный просмотр*. Если мнемосхема не умещается на одну страницу или занимает малую часть страницы, её можно уменьшить нажатием на кнопку  или увеличить нажатием на кнопку . После подготовки мнемосхемы её можно распечатать нажатием на клавишу *Печать...*. Закрывать окно просмотра можно нажатием клавиши *Закреть*.

Для окончания работы программы ГД необходимо выбрать из системного меню пункт меню *Файл*, а в нём пункт *Выход*. На экране появится запрос-предупреждение о сохранении проекта. Нажатие на кнопку *Да* приведёт к выходу из программы с сохранением всех изменений в проекте, сделанных после последнего сохранения проекта. Нажатие кнопки *Нет* приведёт к выходу из программы без сохранения всех изменений в проекте, сделанных после последнего сохранения проекта. Нажатие кнопки *Отмена* приведёт к выходу из данного подменю и не приведёт к выходу из программы ГД.

Подменю Правка. При входе в системное меню пункт *Правка* можно: скопировать, вырезать, вставить и удалить элементы мнемосхемы проекта (рис. 3.29).

Для выделенного элемента можно вызвать *Окно свойств*, выбрав пункт меню *Свойства*.

Для выделения конкретных элементов мнемосхемы необходимо выбрать из системного меню пункт *Правка*, а в нём пункт *Выделить* (рис. 3.30). На экране появится окно с элементами мнемосхемы, доступными для выделения. При необходимости выделить все элементы мнемосхемы нужно выбрать пункт *Выделить все*.

Кроме этого в системном меню пункта *Правка* можно отменить или вернуть действия, произведённые над элементами мнемосхемы.

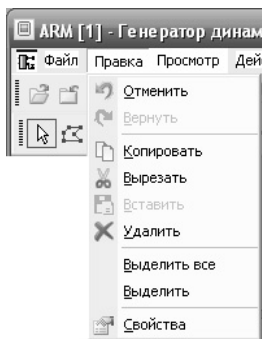


Рис. 3.29. Меню *Правка*

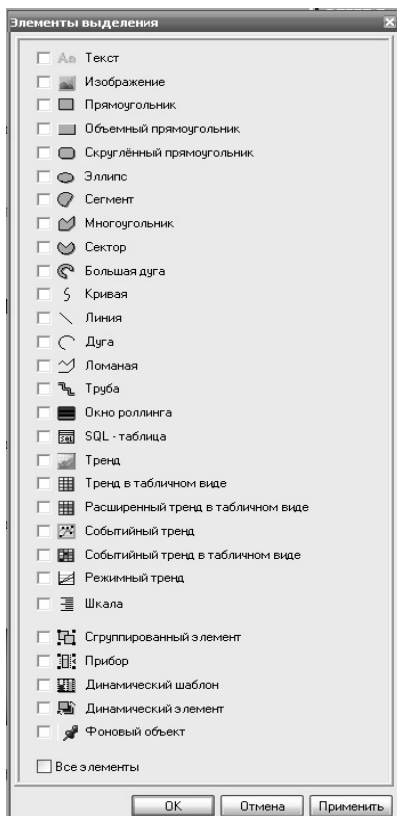


Рис. 3.30. Элементы выделения мнемосхемы

Подменю Просмотр. Подменю *Просмотр* служит для изменения вида панели инструментов, вызова на экран окна проекта, окна свойств, окна библиотеки и включения/выключения строки состояния (рис. 3.31).

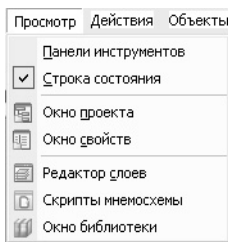


Рис. 3.31. Подменю Просмотр

При выборе пункта подменю *Панели Инструментов* на экране появится подменю (рис. 3.32). В данном подменю можно включить/выключить панель инструментов из списка или создать новую панель.

Для включения панели инструментов необходимо установить курсор в квадрат, расположенный слева от включаемой панели, и нажать левую клавишу мыши. Нажатие в квадрате с отметкой приводит к выключению выбранной панели. Включение/выключение панели сразу видно в стандартной панели инструментов.

Из подменю *Просмотр Главного Меню* можно включить/выключить *Строку Состояния*. *Строка Состояния* появляется в нижней части экрана и служит для вывода более полной подсказки о действии, выполняемом кнопкой или функцией, на которой стоит курсор.

Из подменю *Просмотр* можно включить/выключить *Окно Проекта*. *Окно Проекта* включает в себя два подменю *Объекты* (Мнемосхемы, Рабочие столы, Переходы) и *Библиотеки*. В подменю *Объекты* выдается информация об открытом проекте и включаемых в него объектах. В подменю *Библиотеки* показываются созданные библиотеки, которые могут быть открыты для работы. Для увеличения размера окна мнемосхемы можно закрыть *Окно Проекта*.

Из подменю *Просмотр* можно включить/выключить *Окно Свойств*. Разница в вызове этого подменю из подменю *Правка* и подменю *Просмотр* состоит в том, что из подменю *Правка* можно вызвать *Окно Свойств* только для выделенного элемента, а из подменю *Просмотр* можно вызвать *Окно Свойств* и без выделения элемента.

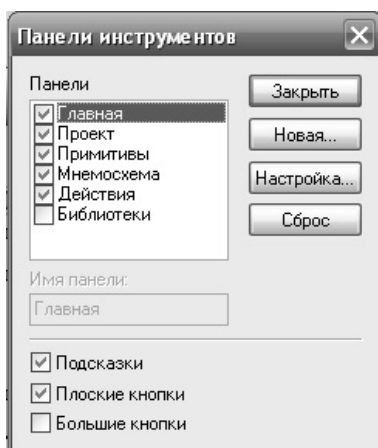


Рис. 3.32. Подменю Панели Инструментов

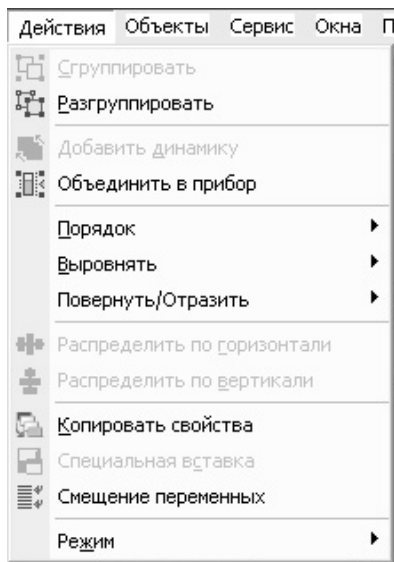


Рис. 3.33. Подменю Действия

Подменю Действия. Подменю *Действия* служит для изменения свойств одного или нескольких выделенных элементов, а также просмотра установленных динамик при изменении режима работы ГД (рис. 3.33).

Пункты подменю, имеющие справа треугольный указатель, имеют свои подменю, вызвать которые можно, установив курсор на нужный пункт.

Группировать – данный пункт меню становится доступным при выделении двух и более элементов мнемосхемы. Сгруппированный объект имеет общие для включённых в него элементов свойства.

Разгруппировать – данный пункт меню становится доступным при выделении сгруппированного объекта мнемосхемы. Сгруппированный объект распадётся на отдельные включённые в него элементы. Если выделен элемент динамики, то нажатие данной кнопки приведёт к отмене динамики.

Добавить динамику – можно любому элементу, созданному на мнемосхеме. Данный пункт меню становится доступным при выделении элемента мнемосхемы. Для элемента, которому добавили динамику, в *Окне свойств* появляются две дополнительные закладки: *Привязка* и *Реакции*.

Для назначения динамики можно выбрать любое свойство из списка закладки *Привязка* для данного элемента. Для каждого элемента имеется свой список атрибутов, которым может быть назначена динамика. Пример назначения динамики выделенному элементу приведён на рис. 3.34.

Если динамика ещё не назначалась, то во всех строках списка будет стоять сообщение **Нет**, говорящее о том, что динамика данному свойству не назначалась. При выборе какого-либо свойства строка будет выделена цветом и в ней появится треугольник, при нажатии на который появится список типов динамик, которые можно назначить данному свойству. Для каждого свойства существует конкретный список типов динамик.

У элемента, которому назначена динамика, в меню свойств появляется ещё одна закладка – *Реакции* (рис. 3.34). Реакции – это отклик системы на нажатие/отжатие левой или правой клавиши мыши, нажатие/отжатие описанных клавиш на выделенном элементе или на вызванной мнемосхеме, попадание курсора в поле объекта или выход курсора за его пределы, последовательный ввод символов. Откликом может быть квитирование звукового сигнала, вызов или активизация окна ввода какого-либо атрибута переменной, изменение состояния логического параметра, ввод какой-либо управляющей команды, переход на другую мнемосхему или рабочий стол, вызов другой мнемосхемы, вызов окна свойств переменной и т.д.

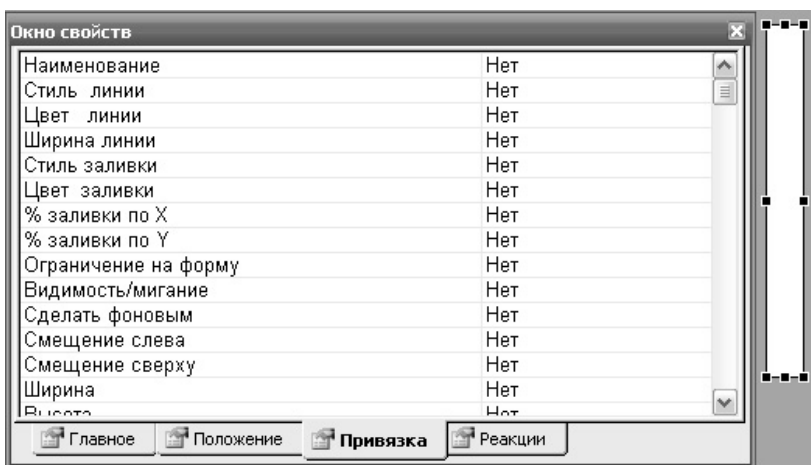


Рис. 3.34. Пример назначения динамики выделенному элементу

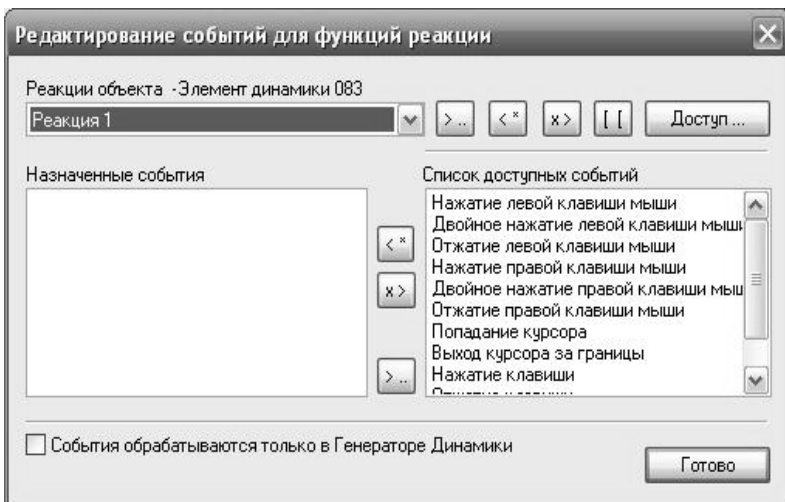



Рис. 3.35. Диалоговое окно Редактирование событий для функций реакции

Прежде чем перейти к описанию реакций, необходимо назначить сами функции. Назначение функций реакции производится по нажатию кнопки  – добавить реакцию. При этом появляется диалоговое окно *Редактирование событий для функций реакции* (рис. 3.35), в котором уже создана реакция, которую необходимо описать.

Функции реакции – это последовательность действий, автоматически выполняемых системой в ответ на какое-либо определённое событие. Таким событием может быть действие пользователя, например, нажатие клавиши клавиатуры, нажатие кнопки мыши или определённая последовательность аналогичных действий.

Перечень функций реакции:

- установить значение;
- поле ввода;
- поле ввода с клавиатурой;
- локальный переход;
- смещение в группе;
- вращение;
- прямой переход;
- квитиование;
- таблица настройки переменной;

- дискретное управление;
- настройка тренда;
- настройка таблицы тренда;
- выполнить программу;
- закрытие мнемосхемы/программы;
- зависимый переход;
- выполнить функцию;
- активизация объекта;
- больше – меньше;
- динамическое изменение входов шаблона;
- групповое управление;
- удаление элемента.

Объединить в прибор можно один или несколько графических элементов и элементов, имеющих динамику. Данный пункт меню становится доступным при выделении элемента мнемосхемы. Для элемента, которому добавили пункт *Объединить в прибор*, в *Окне свойств* присутствуют два пункта: *Главное* и *Переменные*. В пункте меню *Главное* настраиваются размеры и место установки прибора. В пункте меню *Переменные* приводятся все переменные, на которые есть ссылки во всех входящих в прибор элементах динамики (привязках и реакциях). При копировании такого прибора его можно легко настроить, переназначив входящие в него переменные.

Порядок. С помощью команд данного пункта меню можно перемещать элемент на передний или задний план относительно других элементов мнемосхемы.

Выровнять. С помощью команд данного пункта меню можно изменять местоположение и размеры выбранных элементов.

Повернуть/Отразить. С помощью инструментов данного пункта меню можно поворачивать по кругу или зеркально отображать координаты элемента относительно оси элемента или оси группы элементов, если элемент входит в сгруппированный элемент.

Распределить по горизонтали. С помощью данного инструмента вертикальные оси всех выбранных элементов располагаются на одинаковом расстоянии друг от друга. Инструмент становится доступным при выделении трёх и более элементов мнемосхемы.

Распределить по вертикали. С помощью данного инструмента горизонтальные оси всех выбранных элементов располагаются на одинаковом расстоянии друг от друга. Инструмент становится доступным при выделении трёх и более элементов мнемосхемы.

Копировать свойства. С помощью данного инструмента можно копировать выбранные свойства активного элемента для всех выделенных элементов. Для копирования свойств необходимо выделить все элементы, для которых необходимо присвоить одинаковые свойства, причём последним элементом выделить тот, свойства которого присваиваются другим.

Специальная вставка. С помощью данного инструмента можно вставить ранее скопированный элемент, задав смещение по горизонтали и вертикали относительно скопированного элемента. Пользование этим инструментом удобно, когда элементы располагаются в одну линию или в один столбец.

Режим. С помощью данного пункта меню можно изменить режим работы ГД: *Редактирование* и *Имитация*. В режиме *Редактирования* создаются рабочие столы, мнемосхемы, таблицы перехода, т.е. создаётся СИСТЕМА. В режиме *Имитации* производится отладка созданных элементов динамики, проверка назначенных реакций, работы алгоритмов, написанных на языке КРУГОЛ. В этом режиме подключается в работу Сервер БД, поэтому все динамики обновляются в соответствии с работой программы имитации и программы КРУГОЛ, все назначенные реакции посылают в БД назначенные значения.

Подменю объекты. Подменю *Объекты* служит для работы с объектами ГД (рис. 3.36). Все пункты данного меню становятся доступными при открытии окна проекта, где все объекты присутствуют. С помощью пунктов данного подменю можно создать объект или его копию, открыть, закрыть или удалить объект, а также импортировать объект из другого проекта.

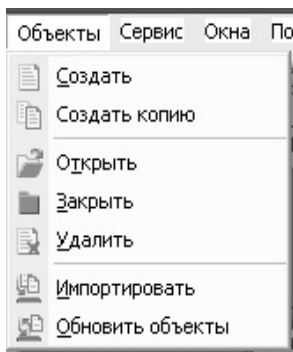


Рис. 3.36. Подменю *Объекты*

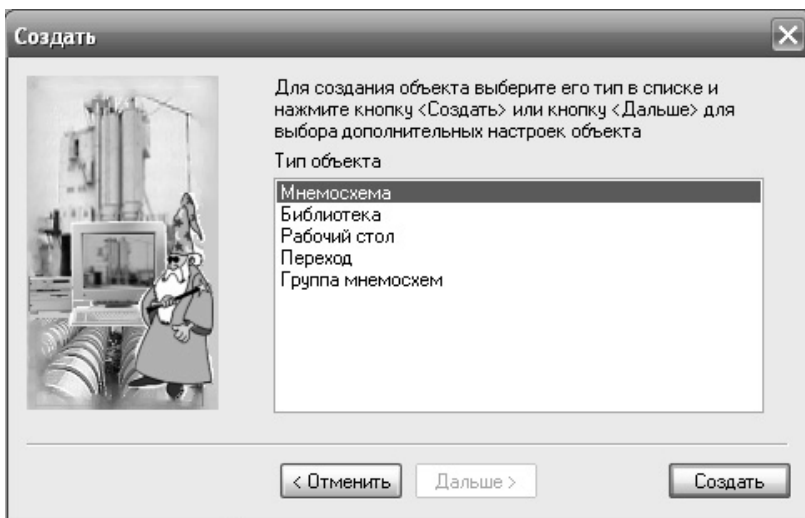


Рис. 3.37. Окно создания объекта проекта в ГД

Создать. С помощью данного пункта меню можно создать объект проекта в ГД. При выборе данного пункта меню появляется окно диалога *Создать* (рис. 3.37).

Для выбора объекта необходимо щелкнуть левой клавишей мыши в строке объекта и далее нажать кнопку *Создать*. После исчезновения окна диалога в *Окне Проекта* появляется объект с порядковым номером, который можно переименовать и назначить ему свойства в *Окне Свойств*.

Создать копию. С помощью данного пункта меню можно создать копию существующего в проекте объекта.

Открыть. С помощью данного пункта меню можно открыть существующий в проекте объект. Для этого необходимо войти в *Окно Проекта*, выделить объект, выбрать пункт *Объекты* Главного Меню и в нём пункт *Открыть*.

Закрыть. С помощью данного пункта меню можно закрыть открытый объект. Для этого необходимо войти в *Окно Проекта*, выделить объект, выбрать пункт *Объекты* Главного Меню и в нём пункт *Закрыть*.

Удалить. С помощью данного пункта меню можно удалить выделенный в *Окне Проекта* объект. Для этого необходимо войти в *Окно Проекта*, выделить объект, выбрать пункт *Объекты* Главного Меню и в нём пункт *Удалить*.

Импортировать. С помощью данного пункта меню можно создать в данном проекте копию существующего в другом проекте объекта.

Обновить. С помощью данного пункта меню можно обновлять в данном проекте мнемосхемы и шаблоны, которые изменялись в другом проекте или на другом компьютере.

Подменю сервис. Подменю *Сервис* служит для включения/выключения сетки, модификаторов, центров и для изменения расположения, количества панелей и кнопок в них в стандартной панели инструментов ГД (рис. 3.38).

Показать сетку. Данный пункт подменю служит для включения/отключения сетки. Сетка служит для того, чтобы по ней легче было устанавливать элементы мнемосхемы, а также для ускоренного перемещения элементов по мнемосхеме и ускорения изменения размеров элементов, так как при одном нажатии на клавиши управления курсором перемещение (изменение размера) происходит на один элемент сетки. Шаг сетки задается в свойствах мнемосхемы и выводится в *Окне Свойств* при выделении имени мнемосхемы в *Окне Проекта*.

Показать выделение. Данный пункт подменю служит для включения/отключения изображения модификаторов элементов. Отключение изображения модификаторов бывает удобным при работе с мелкими элементами или для точной установки элемента. При отключении изображения модификаторов принципы работы с элементами полностью сохраняются.

Показать центры. Данный пункт подменю служит для включения/отключения изображения центров элементов. Включение изображения центров бывает удобным при расположении элементов друг относительно друга.

Настройка. Данный пункт подменю служит для включения/отключения панелей инструментов, а также для настройки панелей – помещения в них дополнительных инструментов из подменю *Группа команд* (рис. 3.39).

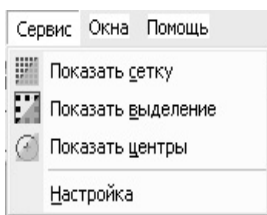


Рис. 3.38. Подменю Сервис

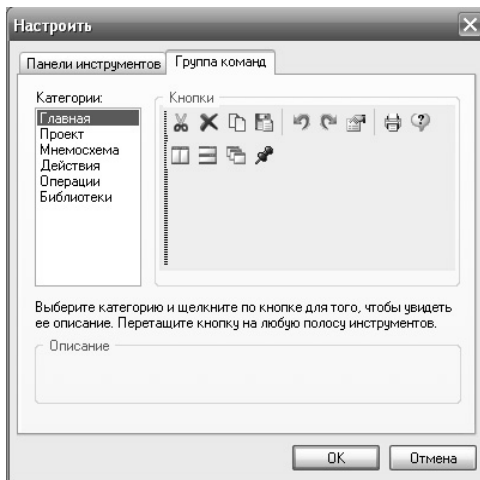
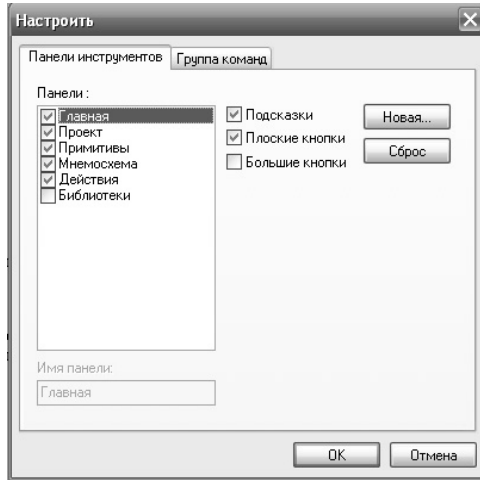


Рис. 3.39. Окно Настройки

Для вызова *Окна Настройки* необходимо выбрать пункт *Настройка* в данном подменю. *Окно Настройки* имеет две закладки. Каждая закладка имеет своё диалоговое окно. В диалоговом окне *Панели инструментов* можно включить/отключить или добавить новую панель инструментов.

В диалоговом окне *Группа команд* можно добавить инструмент в любую *Панель инструментов*.

Подменю окна. Подменю *Окна* служит для переключения между окнами мнемосхем в случае открытия для редактирования нескольких мнемосхем или расположении всех открытых мнемосхем в различном виде при работе ГД (рис. 3.40). В нижней части подменю располагается список открытых мнемосхем. Окно мнемосхемы, в строке которого стоит , является активным.

Каскад. Данный пункт подменю служит для расположения открытых мнемосхем таким образом, чтобы были видны все *Строки Заголовков* окон открытых мнемосхем, как показано в изображении кнопки *Каскад*. Окна при этом принимают уменьшенный размер.

Горизонтальная черепица. Данный пункт подменю служит для расположения открытых мнемосхем таким образом, что *Главное Окно* делится на равные части, в которых располагаются все окна открытых мнемосхем, как показано в изображении кнопки *Горизонтальная черепица*. Окна при этом принимают уменьшенный размер.

Вертикальная черепица. Данный пункт подменю служит для расположения открытых мнемосхем таким образом, что *Главное Окно* делится на равные части, в которых располагаются все окна открытых мнемосхем, как показано в изображении кнопки *Вертикальная черепица*. Окна при этом принимают уменьшенный размер.

Упорядочить значки. Данный пункт подменю служит для того, чтобы упорядочить минимизированные окна открытых мнемосхем в нижней части *Главного Окна*.

Подменю Помощь. Подменю *Помощь* служит для вызова помощи, справки или сведений о программе ГД (рис. 3.41).

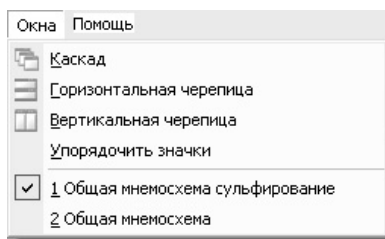


Рис. 3.40. Подменю Окна

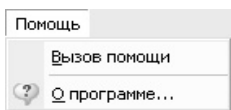


Рис. 3.41. Подменю Помощь

При вызове помощи на экране появится окно *Справка: Генератор динамики*. В закладке *Содержание* представлен список разделов. При выборе какого-либо раздела произойдет переход на описание данного раздела, если у него нет подразделов, или откроется список подразделов. Раскрытый раздел изображается в виде раскрытой книги.

При выборе закладки *Поиск* на экране появится окно *Настройка поиска*. При первоначальном обращении к данному пункту необходимо создать базу данных, содержащую слова справки. С помощью данного окна можно производить поиск помощи по ключевому слову.

3.4. ТЕХНОЛОГИЧЕСКИЙ ЯЗЫК ПРОГРАММИРОВАНИЯ СИСТЕМЫ КРУГ-2000

Интегрированная среда разработки КРУГОЛ является одним из модулей Системы КРУГ-2000. ИСР КРУГОЛ это набор инструментальных средств автоматизации программирования, позволяющих в полном объеме реализовать задачи практически любого уровня сложности и специфики конкретного технологического процесса.

Основными компонентами среды разработки КРУГОЛ являются:

- редактор языка Структурированного текста (СТ);
- редактор языка Функциональных Блочных Диаграмм (ФБД);
- компилятор;
- отладчик;
- библиотека алгоритмов управления и обработки данных.

3.4.1. ОСНОВНЫЕ ПРАВИЛА И ПОНЯТИЯ ТЕХНОЛОГИЧЕСКОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ – ЯЗЫКА СТ

Строка программы может содержать одну из следующих структур языка:

- оператор;
- инструкция;
- оператор + инструкция;
- скобки {или} ограничивающие составную инструкцию оператора;

- комментарий к тексту программы начинаемый с двоеточия.

Операнд – переменная (<тип> <номер>) или константа.

Операция – определитель действия над операндом.

Оператор – определитель изменения порядка выполнения инструкций.

Функция – настройка и реализации сложной задачи одним действием. Применяется только с операцией присвоения.

Инструкция – описатель операций (операции) над операндами (операндом) или функцией. Все операнды математического выражения должны быть одного формата.

Процедура – основная структурная единица программы, содержащая неограниченное число операторов и инструкций. Безусловно выполняется первая процедура по тексту программы.

Программа выполняется построчно сверху вниз.

Операнды технологического языка. Технологический язык КРУГОЛ позволяет описывать разнообразные действия над операндами (переменными различных типов и константами).

Для обозначения переменной в тексте, т.е. её текущего значения, используется конструкция <тип> <номер>. Имеются следующие тринадцать типов переменных, которые охватывают системные переменные, описанные в БД системы (см. ниже под номерами с 1 по 5) и «внутренние» переменные языка (с 6 по 11):

1. **ваN** – текущее значение входной аналоговой переменной;
 2. **авN** – текущее значение выходной аналоговой переменной;
 3. **рвN** – текущее значение переменной ручного ввода (в контроллере используются переменные только вещественного типа);
 4. **вдN** – текущее значение входной дискретной переменной;
 5. **двN** – текущее значение выходной дискретной переменной;
 6. **пвN** – значение «внутренней» переменной вещественного формата;
 7. **пцN** – значение «внутренней» переменной целого формата;
 8. **плN** – значение «внутренней» переменной логического формата (0/1);
 9. **тсN** – секундный таймер – Переменная Вещественного формата;
 10. **тмN** – минутный таймер – Переменная Вещественного формата;
 11. **тчN** – часовой таймер – Переменная Вещественного формата.
- Здесь N – порядковый номер переменной данного типа.

Количество системных переменных (их максимальный порядковый номер) определяется их количеством в БД станции оператора. Диапазон разрешенных номеров для «внутренних» переменных языка типа:

пв, пц – с 1 по 999;

пл – с 1 по X999, где X – количество каналов связи для системы;

тс, тм, тч – с 1 по 25.

«Внутренняя» переменная – это переменная, которую можно использовать, например, для хранения промежуточных результатов вычисления.

Примеры написания операндов в тексте программы:

- переменных: ВА1 ва10 дв13 рв247 пц3 тм2 и т.д.
- констант:
 - логического формата 1 0;
 - целого формата 0 24 -8 и т.д.;
 - вещественного формата 0.15 59.0 – 367.11 и т.д.;

Переменные ВА, АВ, ВД, ДВ, РВ, кроме того, имеют паспорт (атрибуты). Для их обозначения в тексте программы используется конструкция:

<тип><номер>.а<номер атрибута>

где а – латинская или русская буква.

Пример написания операндов с использованием атрибутов:

ВА1.а11, АВ2.а22.

Пример: Изменить программно задание регулятору

Если АВ2.а96 = 0 : Если «Режим ввода задания внешний» = 0
{АВ2.а21 = 29.5} : Присвоить величине задания значение 29.5

Дискретные (логические) переменные. К дискретным (логическим) переменным относятся переменные типа – вд, рв, дв, пл.

Дискретные переменные принимают одно из двух значений: 0 или 1. Все типы дискретных переменных эквивалентны, т.е. допускается сравнивать их между собой, присваивать значение одного типа переменной другому и т.д.

Переменные вд, рв, дв являются системными, т.е. эти переменные описываются в БД системы, логические переменные ручного ввода могут использоваться только на станции оператора.

Целые переменные. К целым переменным относятся переменные типа – пц. Переменная пц используется как внутренняя (промежуточная) переменная, её можно использовать, например, для хранения промежуточных результатов вычисления.

Аналоговые (вещественные) переменные. К аналоговым (вещественным) переменным относятся переменные типов – ва, рв, ав, пв. Переменные ва, рв, ав являются системными, т.е. эти переменные описываются в БД системы.

Таймеры. К таймерам времени относятся переменные типа – тс, тм, тч. Все таймеры – это переменные вещественного формата, значения которых изменяются во времени в соответствии с их типом и могут участвовать в формулах, условиях и т.д. наряду с аналоговыми переменными.

Примеры форматов представления времени:

тс1=20.50 значение таймера № 1 равно 20,5 сек.

тм3=30.50 значение таймера № 3 равно 30 мин. 30 сек.

тч4=5.75 значение таймера № 4 равно 5 час. 45 мин.

Пример использования таймера:

Если тс1 > 20.0 : <Инструкции> будут выполняться
с периодичностью
: 10 сек + время задержки следующего
прохода программы

{
тс1=тс1 – 20.0 : Учёт запаздывания для следующего
десяти секундного
: такта.

<Инструкции>
}

Операции технологического языка

Арифметические операции. Для обработки операндов целого и вещественного формата в языке «КРУГОЛ» реализованы следующие арифметические операции:

Операция	Символ в тексте программы
операция изменения знака	–
операция сложения	+
операция вычитания	–
операция умножения	*
операция деления	/
операция присвоения	=

Логические операции. В языке КРУГОЛ используются следующие логические операции (приведены в порядке их выполнения при использовании в одной инструкции программы):

Операция	Символ в тексте программы
логическое И	&
логическое ИЛИ	
операция присвоения	=

Операции отношения. В языке КРУГОЛ используются следующие операции отношения:

Операция	Символ в тексте программы
больше	>
меньше	<
равно	=
не равно	#

Операции с таймерами.

Вкл – включить таймер временных интервалов.

Выкл – выключить таймер временных интервалов.

Пример: Вкл тм1 – включить минутный таймер номер 1;

Выкл тч2 – выключить часовой таймер номер 2.

Таймеры начинают счёт времени после включения и прекращают счёт после выключения (но не сбрасывают своё значение).

Для того чтобы сбросить таймер на «0» необходимо присвоить ему значение, равное «0».

Операторы технологического языка

Оператор условия. Оператор условия выполняет выбор одной из двух ветвей исполнения программы в зависимости от истинности поставленного условия.

Полная форма записи оператора условия:

ЕСЛИ <условие>

{инструкции и операторы} – выполняется при выполнении условия

ИНАЧЕ – необязательная ветвь оператора

{инструкции и операторы} – выполняется при невыполнении условия

<Условие> описывается в следующих вариантах:

– <переменная> <оператор отношения> <операнд>

– <переменная> <оператор отношения> <выражение>

Примеры:

ЕСЛИ ваб > пв2+рв3 : Если значение входной
налоговой (ваб),
: больше суммы промежуточной
вещественной (пв2)
: и переменной ручного ввода (рв3),

	тогда
{ав1= ваб}	: выходной аналоговой (ав1) присвоить значение
	: входной аналоговой (ваб),
ИНАЧЕ	
{ав3= 0.0}	: иначе присвоить «0»

Оператор последовательности. Оператор последовательности выполняет ряд проходов своей составной инструкции, с подстановкой при каждом проходе, взамен конструкции <тип>[<имя списка>], значения переменной указанного типа под очередным номером из указанного списка.

Форма записи оператора последовательности:

Для <имя списка> [<список>] , <имя списка> [<список>]
{<инструкции и операторы>}

Пример: Для i1 [1...4], i2 [15,20,25,30]
{ПВ [i1] = ВА [i2]}

Допустимое число номеров во всех списках программы не более 1024.

Оператор досрочного окончания процедуры. Предназначен для досрочного по условию выхода из выполняемой процедуры (алгоблока) на исходную ветвь программы (за строкой вызова).

Форма записи оператора досрочного окончания процедуры:

ВЫХОД

Данный оператор можно использовать для досрочного выхода из оператора последовательности, если последний описать как процедуру.

Процедура. Программа пользователя описывается как процедура. Каждая процедура должна начинаться текстом:

ПРОЦЕДУРА <имя процедуры>

НАЧАЛО

В качестве исполнительной части процедуры можно использовать любые инструкции и (или) операторы. Исполнительная часть процедуры должна заканчиваться текстом:

ВЫХОД

КОНЕЦ

При выполнении программы безусловно начинает выполняться первая по тексту процедура. Для того чтобы выполнялись остальные процедуры, нужно указать их имена (операторы) в исполняемых строках программы.

Пример. Необходимо разработать процедуру защиты от опустошения и превышения заданного значения уровня жидкости в ёмкости.

Операнды:

вал – уровень жидкости в ёмкости

дв1 – управление насосом

дв2 – управление задвижкой

Процедура Защита

Начало

Если вал>вал.а16 : Если уровень превысил уставку,
отключить насос
: и открыть задвижку

```
{  
  дв1=0  
  дв2=1  
  message(«Уровень превысил уставку») ;  
}
```

Если вал<вал.а15 : Если уровень меньше нижней
аварийной границы,
: включить насос и закрыть
задвижку

```
{  
  дв1=1  
  дв2=0  
  message(«Уровень меньше НАГ») ;  
}
```

Выход

Конец

Функция. Функция – это результат реализации штатных задач языка программирования, исходные данные (аргументы) которым задает пользователь при написании программы. В качестве аргумента может быть указана константа или переменная.

В тексте программы функции вызываются (обозначаются) через свои имена со скобками (), где указывают аргумент (аргументы). Формат значений и порядок аргументов функций при написании программы должен соответствовать ниже изложенным указаниям (в описаниях функций). В качестве аргументов используются операнды (константы или переменные).

Все функции языка разрешено применять при написании программы только с операцией присвоения.

Пример:

пв1 = п(Аргумент 1, ..., Аргумент М),

где п – имя функции.

3.4.2. СРЕДА РАЗРАБОТКИ СТ

Интегрированная среда разработки СТ состоит из редактора и отладчика.

Редактор СТ предназначен для разработки программ пользователя, которые объединяются в проекты. Пользователь может работать в двух режимах: режиме редактирования и в режиме редактирования с возможностью отладки. В режиме редактирования имеется возможность создавать новые проекты и программы или редактировать уже существующие, сохранять и печатать программы, а также транслировать программы или весь проект с целью получения исполняемых модулей.

Отладчик СТ предназначен для отладки программ пользователя, для поиска ошибок, допущенных при программировании. Отладчик позволяет поэтапно выполнить программу пользователя и просмотреть значения всех переменных, участвующих в программе, а также их атрибуты для детальной проработки алгоритма.

Окно редактора приведено на рис. 3.42.

Окно редактора содержит меню, основную панель инструментов, панель трансляции, панель отладки, окно проекта, окно программы и окно вывода сообщений.

В редакторе (СТ) пользователь работает с файлами проекта, который включает в себя набор программ пользователя и файл профиля с настройками пользовательского интерфейса.

В редакторе СТ пользователь может создавать новые проекты, редактировать ранее созданные проекты и содержащиеся в нём программы, добавляя или удаляя из проекта программы.

Порядок разработки проекта в редакторе СТ заключается в следующем:

1. Создаётся новый проект, с указанием места размещения проекта на диске.

2. В проекте создаются или добавляются программы пользователя. Вновь создаваемые программы можно редактировать с использованием меню «Правка», а также с помощью панели инструментов. Набор текста осуществляется также, как и в любом редакторе.

3. Транслирование проекта. После завершения набора текста программ проекта необходимо транслировать (преобразовать) программы пользователя на языке КРУГОЛ в исполняемые модули, которые затем будут запущены либо на станции оператора, либо в контроллере. Редактор СТ позволяет транслировать как отдельный файл из проекта, так и весь проект полностью.

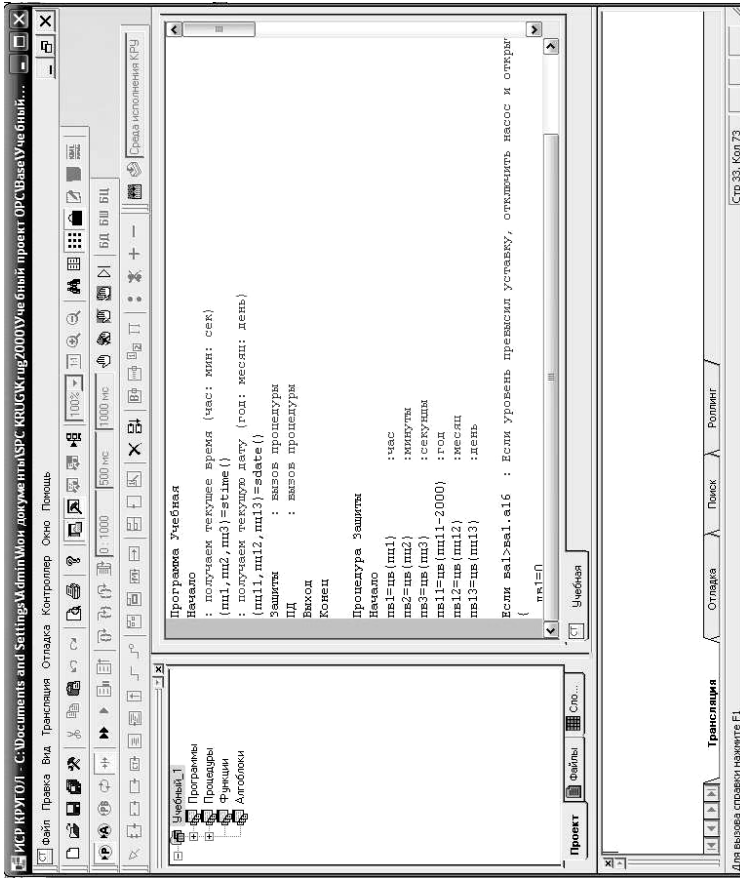


Рис. 3.42. Окно редактора СТ

Перед началом трансляции пользователь должен указать в свойствах проекта (меню «Выполнение») для какой платформы необходимо генерировать исполняемые модули. Платформа, установленная для данного проекта, показывается в информационной строке на панели «Трансляция».

В случае если в ходе трансляции в тексте программы будет обнаружена ошибка, в окне вывода, в зависимости от того какого рода ошибка была обнаружена, появятся соответствующие сообщения. После того как пользователь исправит все ошибки в программе и выполнит трансляцию, в окне вывода появится сообщение об успешной генерации кода.

Интегрированная среда разработки СТ, помимо непосредственного набора, редактирования и трансляции программ, позволяет производить отладку программ пользователя. Отладка подразумевает поэтапное выполнение программ с возможностью по желанию пользователя приостанавливать и возобновлять это процесс. В режиме отладки пользователь может наблюдать за тем, как меняются переменные по ходу работы программы, и по результатам отладки изменять алгоритм, повышая тем самым надёжность и корректность работы программы.

Для работы в режиме отладки служит меню «Отладка», соответствующая ему панель инструментов «Отладка», а также панель «Переменные», служащая для включения/выключения отдельных окон просмотра переменных (рис. 3.42).

3.4.3. СРЕДА РАЗРАБОТКИ ФБД

Интегрированная среда разработки функционально-блочных диаграмм ФБД состоит из редактора и отладчика. Редактор ФБД предназначен для создания и редактирования программ ФБД.

В процессе редактирования схемы редактор ФБД создаёт базу данных программы пользователя, используемую в дальнейшем транслятором. Редактор также предоставляет пользователю набор различных сервисных функций: добавление, выделение, удаление элементов, задание свойств элементам и т.д.

Отладчик ФБД предназначен для отладки программ пользователя, для поиска ошибок, допущенных при программировании. Отладка может осуществляться в циклическом и пошаговом режимах. После выполнения отладки выдается отчёт о результате проверки схемы.

Элементами языка ФБД являются графические символы, используя которые создаётся схема ФБД.

Он позволяет строить сложные процедуры, используя существующие функции из поставляемой библиотеки, и связывать их с другими элементами ФБД. Связь выполняется при помощи специального элемента – связь. Области элемента схемы имеют следующие определения (рис. 3.43).

Элемент «Функция» – это блок, реализующий какую-либо функцию. Может иметь нуль или более входов и нуль или более выходов (рис. 3.44). Входы соединяются с левым краем. Выходы соединяются с правым краем. Каждый вход или выход функции имеет определённый тип, например, – логический, вещественный, целый, строковый.

Элемент «Переменная» – это блок, которому присваивается значение параметра из БД или используется как промежуточная переменная в схеме ФБД (рис. 3.45). Значение этой переменной подаётся либо на вход функции или на вход другой переменной, либо снимается с выхода функции и сохраняется в переменной.

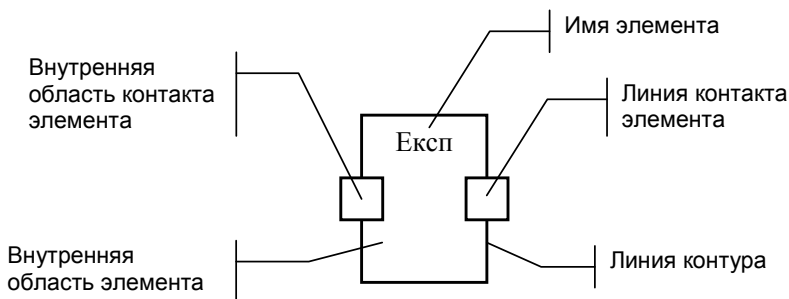


Рис. 3.43. Области элемента схемы ФБД

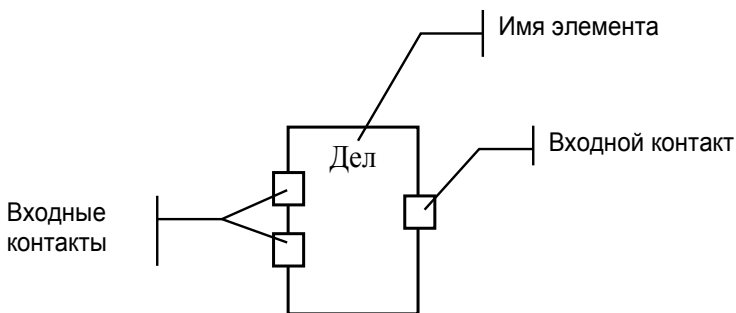


Рис. 3.44. Элемент «Функция»

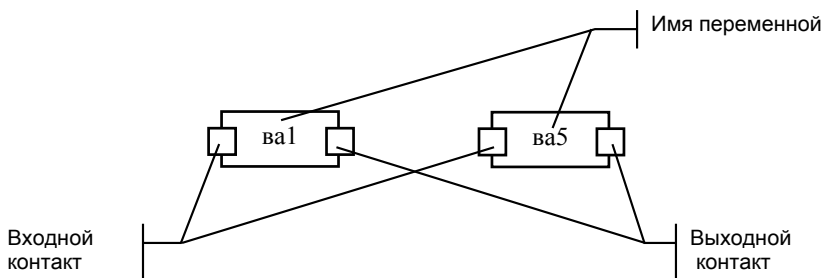


Рис. 3.45. Элемент «Переменная»

Элемент «Константа» – это блок, которому присваивается какое-либо значение пользователя (рис. 3.46). Это значение подается на вход какой-либо функции или переменной.

Элемент «Комментарий» – это блок, содержащий текст пользовательского комментария (рис. 3.47). Комментарий не привязывается к какому-либо конкретному элементу, а свободно располагается где-либо в диаграмме.

Элемент «Связь» – это элемент, устанавливающий зависимость между входами/выходами элементов ФБД (рис. 3.48). Данный элемент может соединять выходы/входы одного типа (логические, вещественные, целые, строковые). Правильность соединений проверяется на этапе назначения.

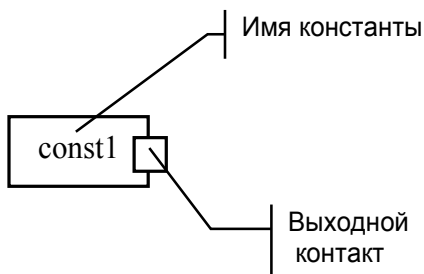


Рис. 3.46. Элемент «Константа»



Рис. 3.47. Элемент «Комментарий»

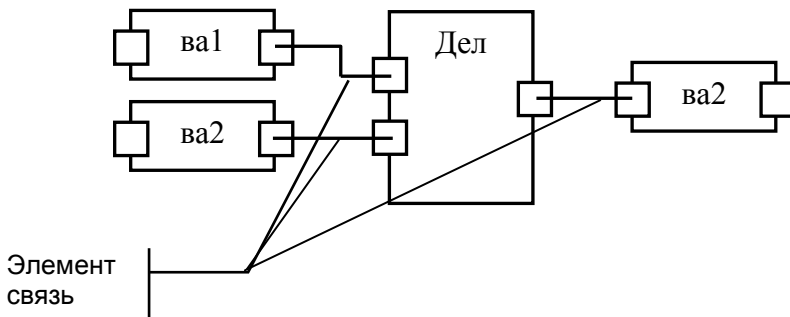


Рис. 3.48. Элемент «Связь»

Элемент языка ФБД оператор «ЕСЛИ» служит для организации ветвления в программе по условию. Элемент имеет один вход логического типа. Если на вход поступает значение 1, то происходит выполнение элементов схемы ФБД в секции <Если>, если 0 – выполняются элементы схемы ФБД в секции <Иначе> (рис. 3.49).

Элемент языка ФБД оператор «ДЛЯ» предназначен для организации циклического выполнения операций схемы ФБД, помещенных в прямоугольник, ограничивающий данный блок (рис. 3.50). Блок имеет один вход логического типа. Если на вход поступает значение 1, блок выполняется, если 0 – не выполняется. Каждый блок «ДЛЯ» должен содержать один или более элементов «Массив данных».

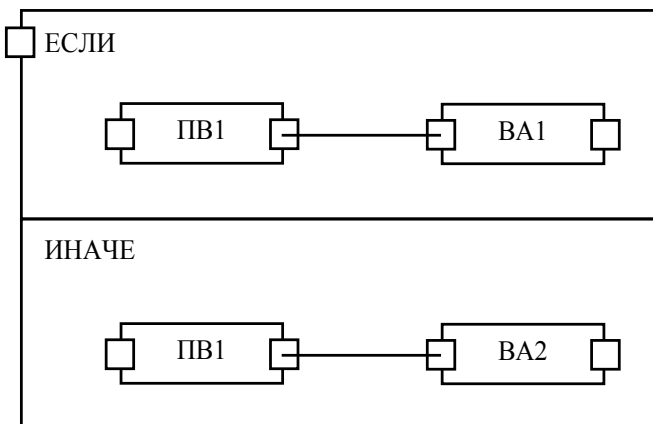


Рис. 3.49. Оператор «ЕСЛИ»

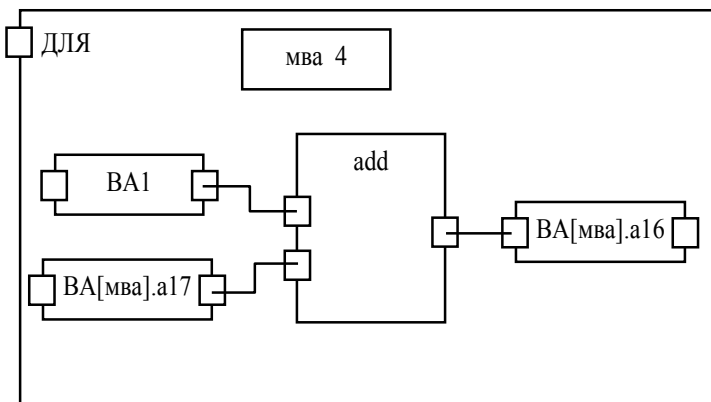


Рис. 3.50. Оператор «ДЛЯ»

Размерность массива определяет число циклов, которое будет выполнено соответствующим оператором «ДЛЯ». Размерность всех массивов данных в пределах данного блока «ДЛЯ» должна быть одинаковой.

Элемент языка ФБД оператор «ВЫХОД» используется для досрочного завершения выполнения программы/процедуры/функции (рис. 3.51).

Элемент имеет один вход логического типа. Выход из программы/процедуры/функции происходит, если на вход поступает значение 1, иначе – выход не происходит.

Элемент ФБД «Процедура» используется для вызова программы ФБД или СТ находящейся во внешнем файле *.lg или *.dfb (рис. 3.52). Блок имеет один вход логического типа. Если на вход поступает значение 1, происходит вызов процедуры, если 0 – вызова процедуры не происходит.

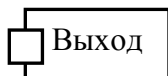


Рис. 3.51. Оператор «ВЫХОД»

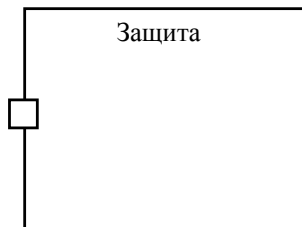


Рис. 3.52. Элемент «Процедура»

3.4.4. СРЕДА РАЗРАБОТКИ ФБД

Интегрированная среда разработки ФБД по аналогии со средой разработки СТ состоит из редактора и отладчика.

В редактор ФБД разрабатываются программы пользователя, которые объединяются в проекты.

Отладчик БФД предназначен для отладки программ пользователя, для поиска ошибок, допущенных при программировании. Отладчик позволяет поэтапно выполнить программу пользователя и просмотреть значения всех переменных, участвующих в программе, а также их атрибуты для детальной проработки алгоритма.

Окно редактора ФБД приведено на рис. 3.53.

Окно редактора содержит меню, основную панель инструментов, панель трансляции, панель отладки, окно проекта, окно программы и окно вывода сообщений.

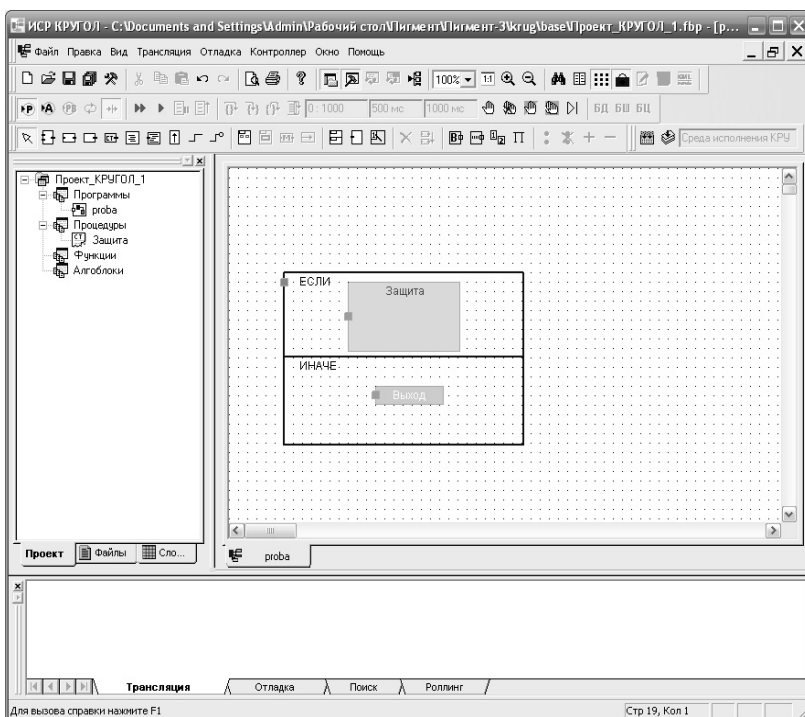


Рис. 3.53. Окно редактора СТ

В редакторе (ФБД) пользователь работает с файлами проекта, который включает в себя набор программ пользователя и файл профиля с настройками пользовательского интерфейса.

В редакторе ФБД пользователь может создавать новые проекты, редактировать ранее созданные проекты и содержащиеся в нём программы, добавляя или удаляя из проекта программы.

Порядок разработки и отладки проекта в редакторе ФБД аналогичен порядку разработки и отладки проекта в редакторе СТ.

3.5. СОЗДАНИЕ ПРОСТЕЙШЕГО ПРОЕКТА

Необходимо разработать автоматизированное рабочее место оператора АСУ ТП производства горячей воды в водогрейном котле (рис. 3.54).

1. Создание базы данных. Запускаем Генератор БД, выбрав в меню ПУСК / Программы / Система КРУГ-2000 / Среда разработки / Генератор базы данных». Создаём новую базу данных, указав путь к директории, в которой будет храниться создаваемая БД и имя файла БД.

Для работы с новой базой данных необходимо щёлкнуть мышью на кнопке «ОК». Основное окно генератора БД появляется сразу после выбора (или создания) файла с БД (рис. 3.54).

Опишем основные поля, необходимые для создания БД.

Выбираем пункт из главного меню Система, в котором выбираем иконку «Общесистемные настройки» и в открывшемся окне вводим название АСУ ТП.

Далее выбираем иконку «Абоненты» где описываем всех абонентов, входящих в структуру ПТК (рис. 3.55).

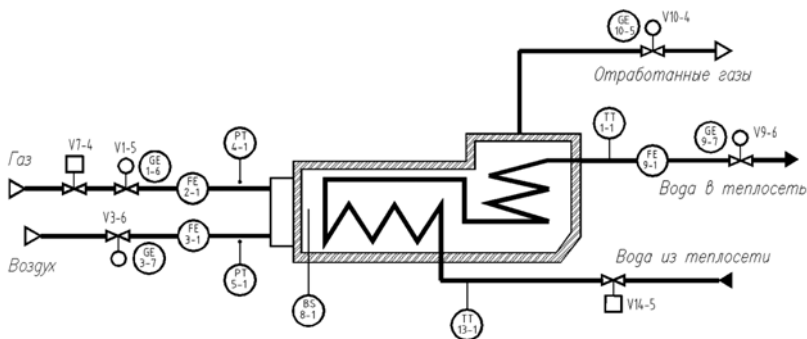


Рис. 3.54. Схема автоматизации водогрейного котла

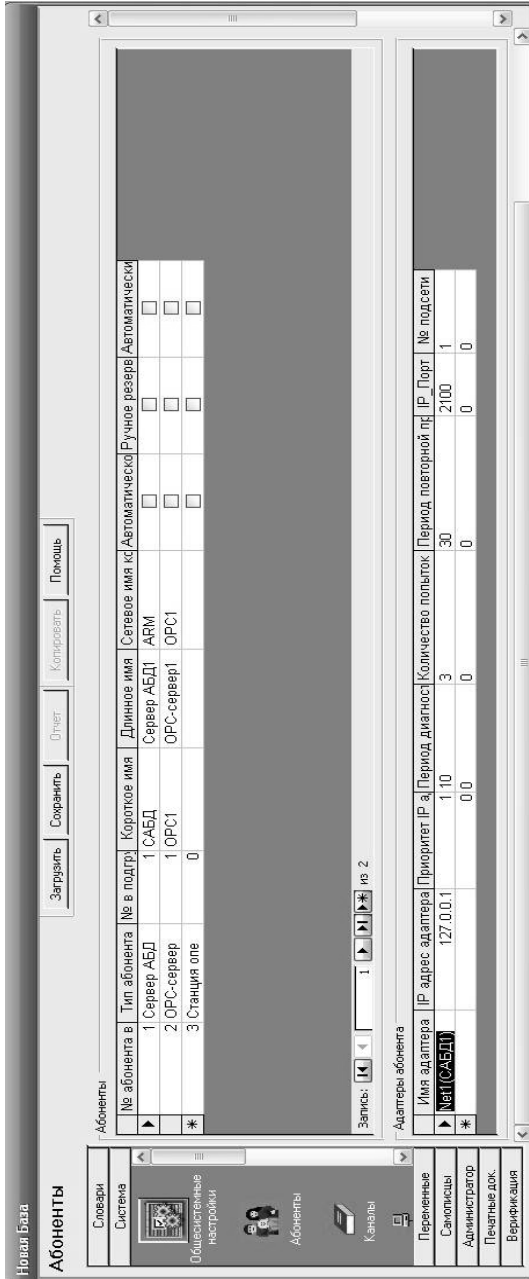


Рис. 3.55. Окно описания абонентов ПТК

Параметры описания абонентов следующие: номер абонента в ПТК – определяет уникальный порядковый номер абонента в данном ПТК; тип абонента – выбирается из выпадающего списка типов абонентов ПТК; короткое имя абонента – пользовательское короткое имя абонента; длинное имя абонента – пользовательское длинное имя абонента; сетевое имя компьютера – имя компьютера, заданное в описании настройки сети для данного абонента.

Описываем адаптеры абонентов (рис. 3.55). Параметры описания абонентов следующие: номер адаптера – определяет уникальный порядковый номер адаптера абонента; имя адаптера – пользовательское имя адаптера, IP адрес адаптера – для абонента «станция оператора» указывается IP адрес компьютера, а для контроллера – 127.0.0.1 порт 2100; Приоритет IP – устанавливается приоритет (например, 1); также указываются период диагностики, количество попыток связи и период повторной диагностики; IP порт – значение порта указываем равным 2100; номер подсети – определяет порядковый номер сети адаптера.

Заполняем вкладку «Каналы» (рис. 3.56). В ней указываются номер и тип канала, по которому будет происходить обмен (например номер – 1, тип – сервер ввода/вывода) и выбирается номера и имена абонентов ПТК из выпадающих меню.

Следующим этапом является заполнение пункта «Переменные». Для создания проекта можно обойтись переменными трёх типов: входные аналоговые, входные дискретные и выходные дискретные (рис. 3.57).

Выбираем иконку «Входная аналоговая» – выводится табличная форма для описания входной аналоговой переменной, в которой заполняем следующие атрибуты: № Переменной – текущий номер входной аналоговой переменной в базе данных сервера БД; № Канала – номер канала, по которому будет происходить обмен данными между сервером БД и УСО; № Переменной в УСО – текущий номер входной аналоговой переменной УСО; позиция – код параметра, в качестве которого обычно используется технологическая позиция данного параметра, которая в дальнейшем может использоваться при отображении; имя1 – это первые 8 символов в названии параметра, которые в дальнейшем можно использовать при отображении; имя2 – это вторые 8 символов в названии параметра, которые в дальнейшем можно использовать при отображении; единица измерения – единица измерения параметра, выбирается из выпадающего списка единиц измерения при его активизации; также нужно указать начало и конец шкалы, а также начало и конец шкалы тренда. Остальные атрибуты заполнять не обязательно.

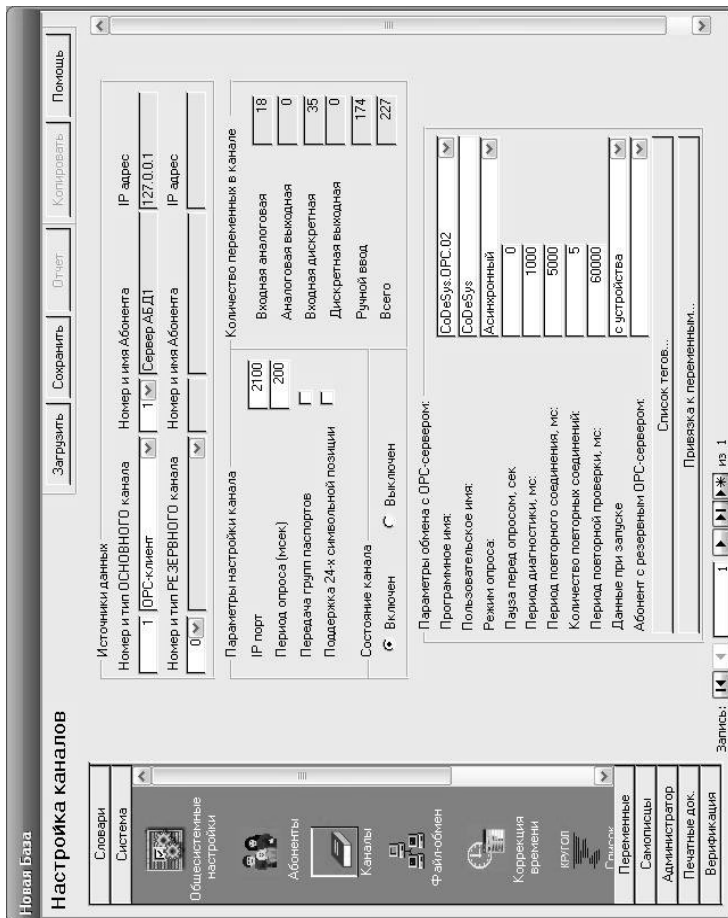


Рис. 3.56. Окно описания каналов

Входная аналоговая														
Загрузить Сравнить Отчет Копировать Поиск														
Словари	№ переменной	№ канала	№ платы	№ входа	№ перем	№ канала	№ УСО в	Основной/Рез	Позиция	Имя 1	Имя 2	Длинная позиция	Длинное имя	Единица изм
Система	1	1	0	0	1	0	0	0	Без резерв	ТТ1	темпер воды			2 - °С
Переменные	2	1	0	0	2	0	0	0	Без резерв	ТТ13	темпер из тепл			2 - °С
	3	1	0	0	3	0	0	0	Без резерв	FE9	расход воды			12 - м3/ча
	4	1	0	0	4	0	0	0	Без резерв	FE3	расход воздух			12 - м3/ча
	5	1	0	0	5	0	0	0	Без резерв	FE2	расход газа			12 - м3/ча
Входная аналоговая	6	1	0	0	6	0	0	0	Без резерв	PT4	давлен газа			8 - кПа
	7	1	0	0	7	0	0	0	Без резерв	PT5	давлен воздух			8 - кПа
	8	1	0	0	8	0	0	0	Без резерв	PT12	давлен воды			8 - кПа
	9	1	0	0	9	0	0	0	Без резерв	PT10	давлен в помп			8 - кПа
Аналоговая входная	10	1	0	0	10	0	0	0	Без резерв	GE1	степени отк. У1-			62 - %
	11	1	0	0	11	0	0	0	Без резерв	GE3	степени отк. У3-			62 - %
	12	1	0	0	12	0	0	0	Без резерв	GE9	степени отк. У9-			62 - %
	13	1	0	0	13	0	0	0	Без резерв	GE10	степени отк.У10			62 - %
Входная дискретная	14	0	0	0	0	0	0	0	Без резерв					

Рис. 3.57. Заполнение пункта «Переменные»

Выбираем иконку «Входная дискретная» – выводится табличная форма для описания входной дискретной переменной, в которой заполняем следующие атрибуты: № Переменной; № Канала; № Переменной в УСО; позиция; имя1; имя2. Остальные атрибуты заполнять не обязательно.

Выбираем иконку «Выходная дискретная» – выводится табличная форма для описания выходной дискретной переменной, в которой заполняем следующие атрибуты: № Переменной; № Канала; № Переменной в УСО; позиция; имя1; имя2. Остальные атрибуты заполнять не обязательно.

Следующим пунктом главного меню, после того, как описаны все типы переменных и конфигурация системы в целом, является Самописцы, которые предназначены для непрерывной регистрации и хранения каких-либо значений параметров системы любых форматов в течение определённого интервала времени. Структурной единицей самописца является «перо самописца» – ссылка на какой-либо атрибут в таблицах переменных системы, значение которого требуется регистрировать с заданным периодом записи. Главное меню состоит из следующих пунктов: базовые самописцы (привязаны к стандартным интервалам времени, например: секунды, минуты, часы, дни и т.д.), базовые перья, производные самописцы и перья производных самописцев. При выборе иконки «Базовые самописцы», на экран вызывается табличная форма для работы со списком базовых самописцев, в которой заполняем следующие поля: № самописца – порядковый номер самописца в системе, каждый самописец должен иметь индивидуальный номер; имя самописца – например: секундный, минутный, часовой; количество перьев – количество перьев, назначенных для данного самописца; кратность записи – периодичность сохранения значений перьев самописцев из оперативной памяти компьютера на жёсткий диск в интервалах записи; период записи – период записи точек самописца в выбранных единицах измерения периода записи; единица записи – признак и единица времени для указанного периода записи точек самописца (выбирается из списка). Выбираем: новая секунда, новая минута, новый час; период обработки – период вызова функции обработки перьев самописца в выбранных единицах измерения периода обработки; единица обработки – признак и единица времени для указанного периода обработки (новая секунда, новая минута, новый час).

Следующим этапом является описание перьев базовых самописцев (рис. 3.58). При выборе иконки «Базовые перья» на экран выводится форма для описания перьев базовых самописцев.

Выбираем автоматический режим заполнения перьев самописцев, который предназначен для быстрого заполнения перьев самописца переменными выбранного типа и выполняется с помощью кнопки «Автозаполнение».

Таким образом, разработали статическую БД. Для создания динамической базы данных необходимо сохранить её, указав путь для сохранения.

2. Создание графического интерфейса. Графический интерфейс создаётся в ГД. Для начала создаётся новый проект (указывается путь и имя) (рис. 3.58). В окне проекта указывается его состав, количество мнемосхем, рабочих столов и переходов.

Сначала необходимо создать новые мнемосхемы (например – рабочий стол (совокупность мнемосхем, при помощи которых организуется внешний вид главного окна программы), заставка и технологический процесс). Характеристики мнемосхем (имя, ширина, высота и т.д.) можно изменять в *Окне свойств*. Далее создаются новые переходы и в таблице переходов указывается, на какие мнемосхемы (например – заставки и технологического процесса) осуществляется переход (рис. 3.59).

После этого создаётся рабочий стол и содержимому рабочего стола назначается мнемосхема «рабочий стол», делается стол основным и указывается переход при открытии стола (например – переход на мнемосхему Заставка). Изменения в рабочий стол вносятся нажатием правой кнопки мыши на рабочий стол и выбором меню «Изменить стол» (рис. 3.60).

После создания проекта необходимо связать его с созданной ранее БД. Для этого в менеджере задач системы КРУГ-2000 создаётся новый проект и в его свойствах указывается его имя, путь к БД и проекту графического интерфейса (рис. 3.61). Дальнейшее изменение графического интерфейса производится из менеджера задач системы КРУГ-2000 нажатием кнопки «Генератор динамики».

Графический интерфейс создаётся с помощью встроенных графических элементов (объёмный прямоугольник; групповой тренд; кривая; изображение; текст и т.д.) и библиотеки шаблонов, которая устанавливается отдельно. Каждый графический элемент имеет свой список свойств, достаточный для описания элемента. Просмотреть список свойств для выделенного элемента (графического примитива) можно в *Окне Свойств*.

Словари Система Переменные Самописцы

Имя самописца

1 секундный

Количество перьев 13

Количество точек 1000

Период записи 1 секунд

Имя пера

Автозаполнение Изменить

Изменить группу перьев Копирование

Удалить

Вид формы

Основная

Переменная

N° пера	Источник данных		Тип обработки		Источник данных по условию	
	Таблица	Запись	Атрибут	Таблица	Запись	Атрибут
1	Перо с П11	1	Текущее значение посл	Среднее за интервал	пусто	пусто
2	Перо с П13	2	Текущее значение посл	Среднее за интервал	пусто	пусто
3	Перо с FE9	3	Текущее значение посл	Среднее за интервал	пусто	пусто
4	Перо с FE3	4	Текущее значение посл	Среднее за интервал	пусто	пусто
5	Перо с FE2	5	Текущее значение посл	Среднее за интервал	пусто	пусто
6	Перо с P14	6	Текущее значение посл	Среднее за интервал	пусто	пусто
7	Перо с P15	7	Текущее значение посл	Среднее за интервал	пусто	пусто
8	Перо с P112	8	Текущее значение посл	Среднее за интервал	пусто	пусто
9	Перо с P110	9	Текущее значение посл	Среднее за интервал	пусто	пусто
10	Перо с BE1	10	Текущее значение посл	Среднее за интервал	пусто	пусто
11	Перо с BE3	11	Текущее значение посл	Среднее за интервал	пусто	пусто
12	Перо с BE9	12	Текущее значение посл	Среднее за интервал	пусто	пусто
13	Перо с BE10	13	Текущее значение посл	Среднее за интервал	пусто	пусто

Базовые самописцы

Базовые перья

Интервалы времени

Произвольные самописцы

Рис. 3.58. Заполнение базовых самописцев

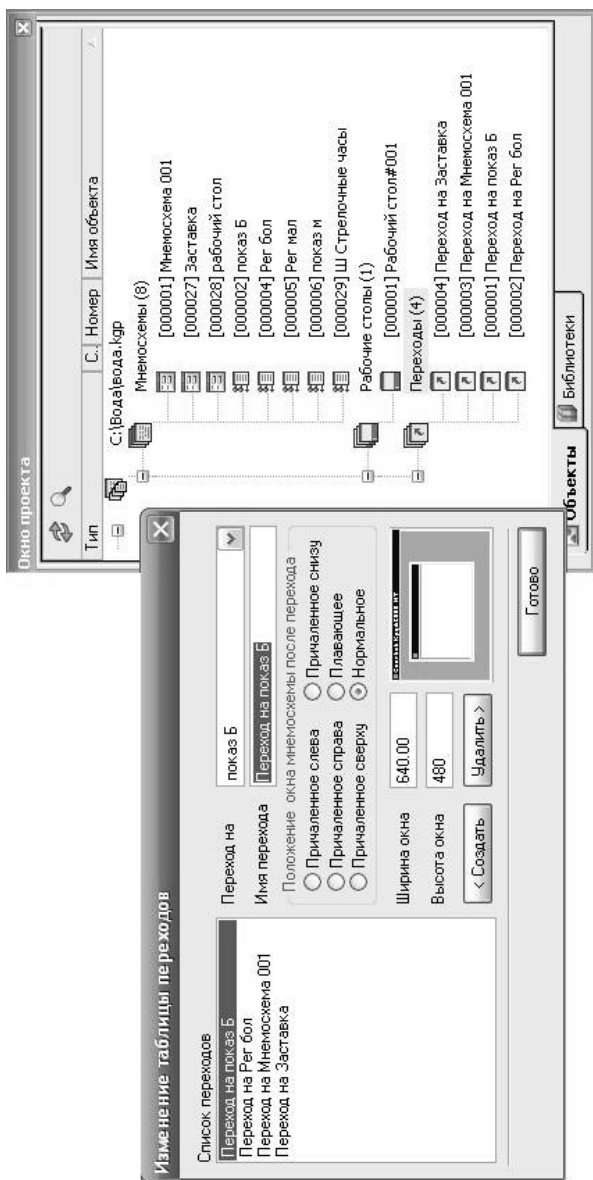


Рис. 3.59. Таблица переходов

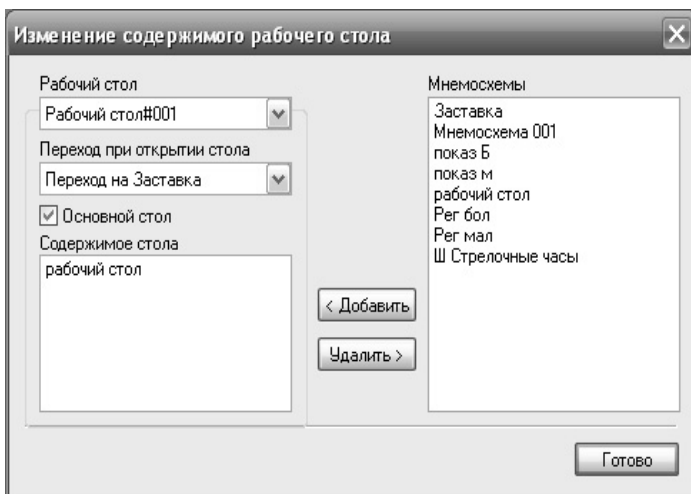


Рис. 3.60. Изменение содержания рабочего стола

Для того чтобы осуществлять переход между мнемосхемами в проекте нужно на мнемосхеме «Рабочий стол» создать кнопки, используя графические примитивы, и наложить на них динамику (рис. 3.62).

После этого в окне свойств данного элемента появится две новых вкладки «Привязка» и «Реакции». Для создания перехода на мнемосхему внутри рабочего стола нужно во вкладке «Реакции» создать новую реакцию (функцию обработки действий пользователя), выбирается тип «Локальный переход» и указывается его имя (рис. 3.63).

После создания мнемосхем и размещения на них необходимой информации (для мнемосхемы технологический процесс – аппаратов, трубопроводов и т.д.) переходят к созданию шаблонов приборов. Как правило, на мнемосхемах размещают приборы, на которых выводится минимум необходимой информации (например – позиция и текущее значение), а при нажатии на них появляются приборы с более подробной информацией (например – позиция, имя, текущее значение, единица измерения, тренд переменной и т.д.). Создание приборов начинается с того, на котором выводится больше информации. Для этого создаётся новая мнемосхема и в окне свойств в пункте «Тип схемы» выбирается «Шаблон прибора» (рис. 3.64).

С помощью графических элементов генератора динамики размещаем на мнемосхеме прибора необходимую информацию (рис. 3.65).

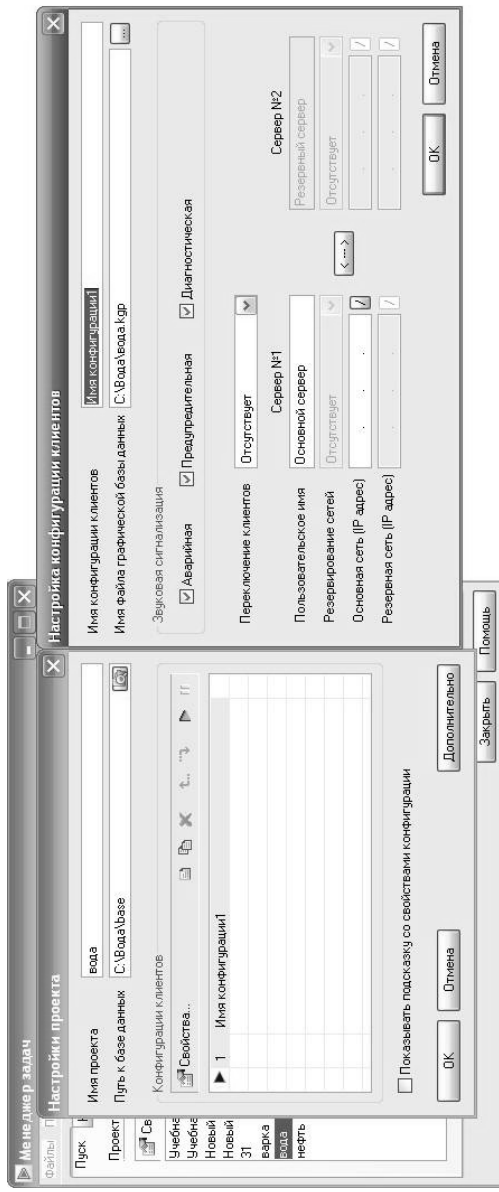


Рис. 3.61. Настройка проекта в менеджере задач системы КРУГ-2000

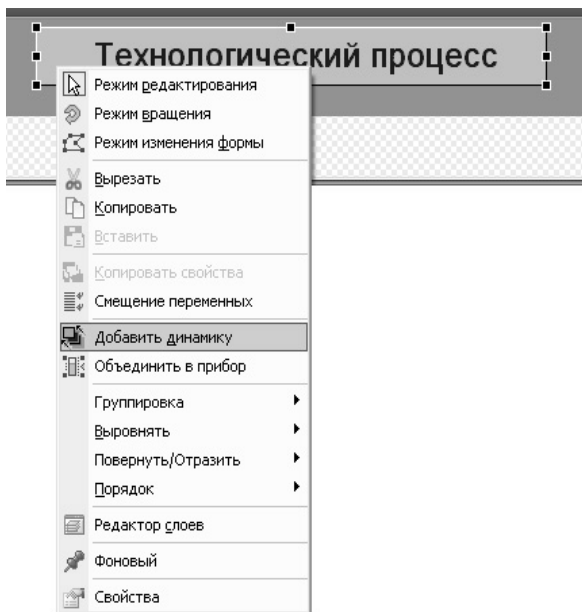


Рис. 3.62. Наложение динамики на графический примитив

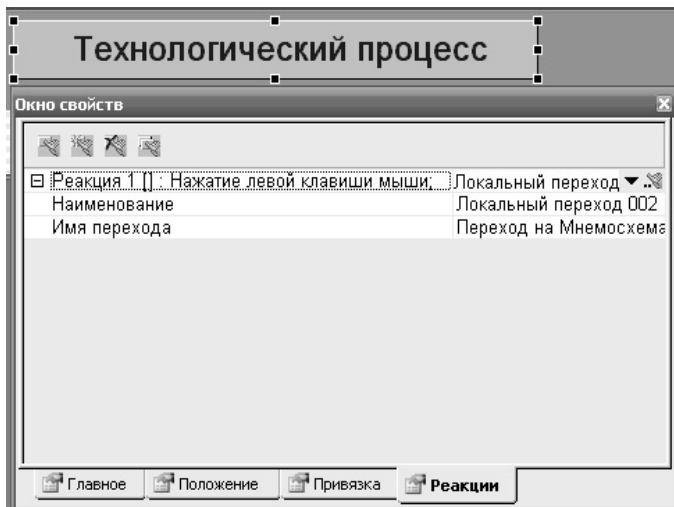


Рис. 3.63. Создание локального перехода

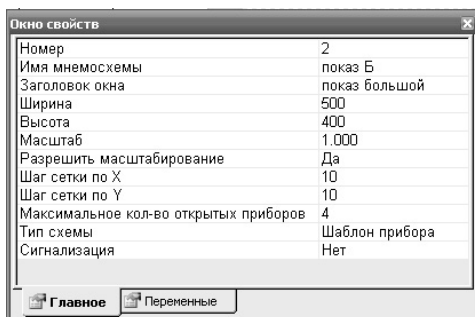


Рис. 3.64. Создание шаблона прибора

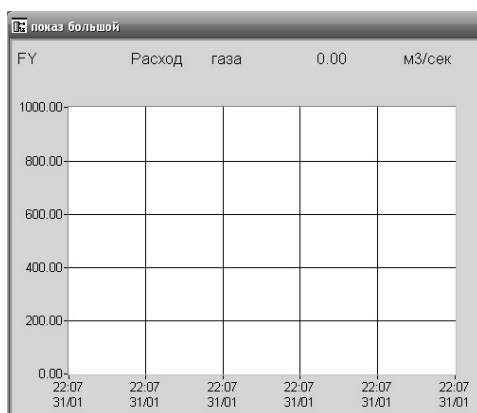


Рис. 3.65. Шаблон прибора

Далее описываются входы прибора. Для этого нажатием правой кнопки мыши в выпадающем меню выбираем пункт «Изменить привязку шаблона» (рис. 3.66). В появившемся окне добавляем входы прибора, выбираем для них тип и ссылку на переменную.

После этого создаём прибор, который размещают на мнемосхеме. Для этого создаётся новая мнемосхема и в окне свойств в пункте «Тип схемы» выбирается «Шаблон прибора». С помощью графических элементов генератора динамики размещаем на мнемосхеме прибора необходимую информацию и на один из графических элементов добавляется реакция «Зависимый переход», где указывается имя перехода и привязка входов прибора (рис. 3.67). После описываются входы прибора.

Далее на мнемосхеме технологического процесса размещаются приборы, в которых происходит привязка переменных к входам прибора.

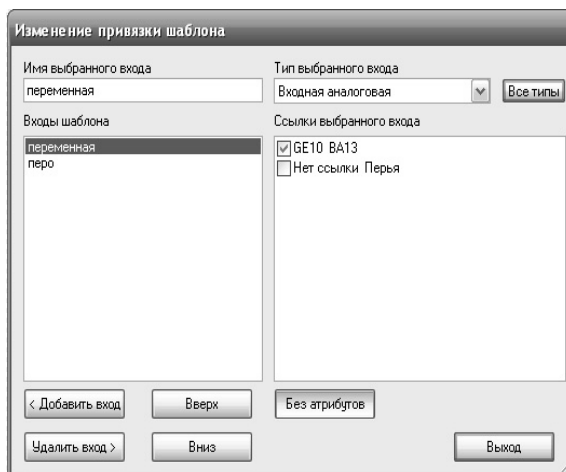
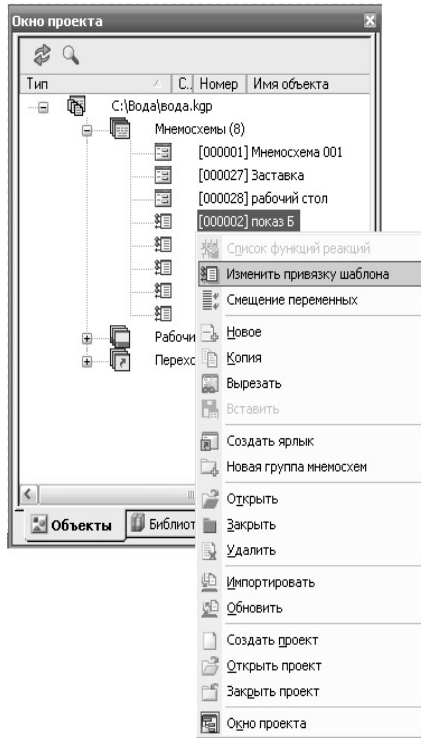


Рис. 3.66. Привязка шаблона прибора

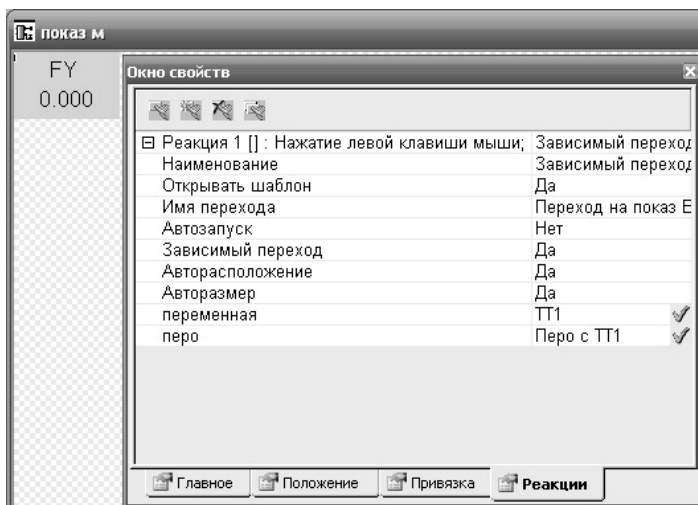


Рис. 3.67. Создание зависимого перехода

3. Настройка OPC-клиента. Настройка OPC-клиента производится запуском Генератора БД. В нём во вкладке «Система» выбирается иконка «Каналы» где происходит непосредственная настройка OPC-клиента (рис. 3.56). В источниках данных задается номер и тип основного канала, через который будет передаваться информация, номер и имя абонента. Далее производится настройка параметров обмена с OPC-сервером где выбирается программное имя OPC-сервера и режим обмена (синхронный или асинхронный).

Затем необходимо определить список тегов контроллера используемых в проекте системы КРУГ-2000. Для этого нажимается кнопка «Список тегов» (рис. 3.56). После этого осуществляется привязка переменных БД системы КРУГ-2000 к переменным (тегам) контроллера. Для этого нажимается кнопка «Привязка к переменным» (рис. 3.68).

4. Настройка «Станции оператора». После создания графического интерфейса в генераторе динамики следует произвести настройку «Станции оператора» в Менеджере задач системы КРУГ-2000. Список требуемых для функционирования задач представлен на рис. 3.69.

После запуска «Станции оператора» загружается графический интерфейс с мнемосхемой технологического процесса (рис. 3.70).

Привязка тегов OPC-сервера (Абонент "1 - САБД")
CoDeSys-OPC.02

Группы тегов Автозаполнение Копирование Удаление Проверка Фильтрация Очистить фильтр

Номер	Имя тега	Имя атрибута	Тип атрибута	Тип п	№ в UI	№ в БД	Позиция пер	Длина	Имя атрибута л	Имя атрибута п	Тип атрибута л	Тип атрибута п	Принем	Передача	Имя гр
1	PLC1_delta_T	Item Value	Вещес PB	0	116	116	deltaT	11	Текущее зн	Вещественн	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
10	PLC1_E_mech	Item Value	Вещес BA	0	13	13	E1E16	Ток дк 28	Текущее зн	Вещественн	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
11	PLC1_F_voda	Item Value	Вещес BA	0	8	8	FQE3	Расхо 28	Текущее зн	Вещественн	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
16	PLC1_in_count_voda	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
17	PLC1_kj_azot_close	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
18	PLC1_kj_azot_control_auto	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
19	PLC1_kj_azot_control_man	Item Value	Логиче PB	0	66	66	V20-3	20	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
20	PLC1_kj_azot_mode	Item Value	Логиче PB	0	67	67	V20-3mode	20	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
21	PLC1_kj_azot_open	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
22	PLC1_kj_azot_SQH	Item Value	Логиче BD	0	20	20	GS20-3	Состо 45	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
23	PLC1_kj_azot_SQL	Item Value	Логиче BD	0	21	21	GS20-4	Состо 45	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
24	PLC1_kj_formalm1_close	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
25	PLC1_kj_formalm1_control_a	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
26	PLC1_kj_formalm1_control_m	Item Value	Логиче PB	0	56	56	V8-4	20	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
27	PLC1_kj_formalm1_mode	Item Value	Логиче PB	0	57	57	V8-4mode	20	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру
28	PLC1_kj_formalm1_open	Item Value	Логиче 0	0	0	0	пусто	пусто 0	пусто	пусто	0	0	<input type="checkbox"/>	<input type="checkbox"/>	0 - Пус
29	PLC1_kj_formalm1_SPH	Item Value	Логиче RP	0	10	10	ASR-5	Состо 15	Текущее зн	Логическое	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 - Гру

Запись: 9 | * из 311

Описание текущего тега
Имя тега PLC1_kj_azot_open

Имя атрибута тега Item Value

Тип атрибута тега Логическое

Параметры текущей привязки
Направление обмена Г
Принем Г
Передача Г
Имя группы тегов

Описание переменной текущей привязки
Имя тега PLC1_kj_azot_open

Тип № в БД № в УСО Позиция переменной Имя атрибута переменной Тип атрибута переменной
пусто 0 0 пусто Имя атрибута переменной Имя атрибута переменной Имя атрибута переменной
Изменить привязку
Сохранить привязку

Рис. 3.68. Привязка к переменным

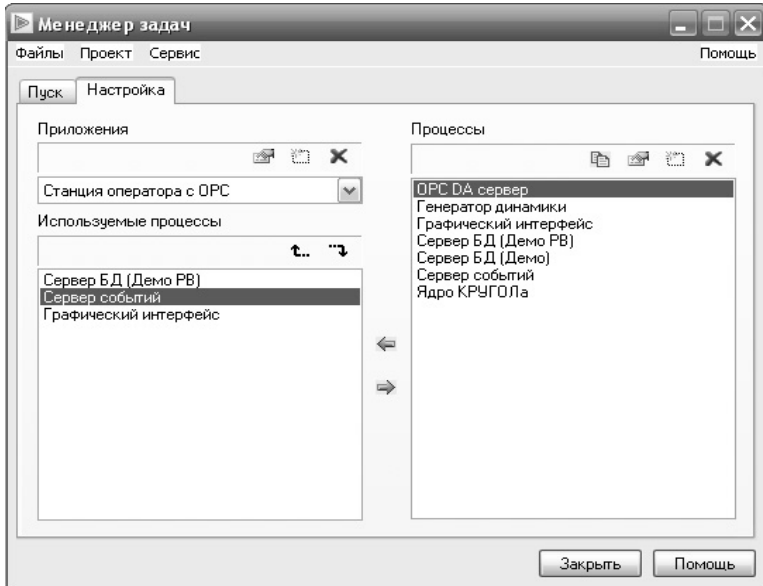


Рис. 3.69. Настройка Станции оператора

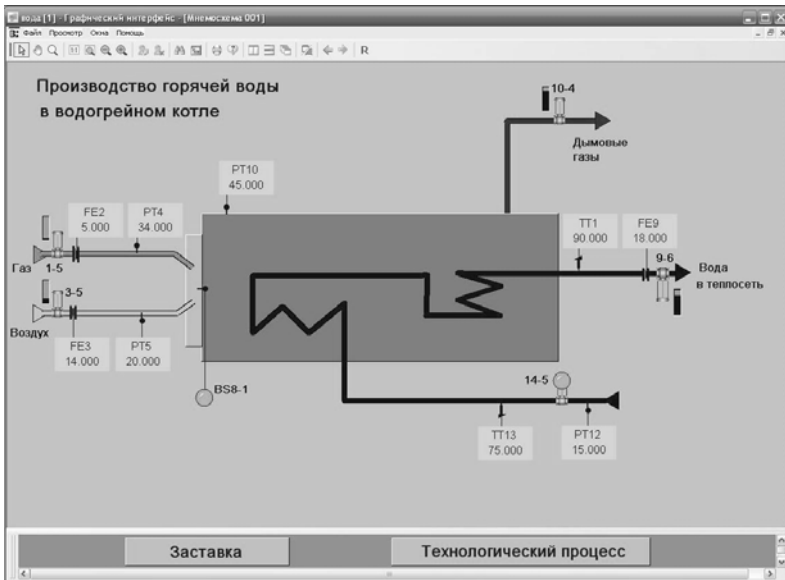


Рис. 3.70. Мнемосхема технологического процесса

3.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие архитектуры АСУТП реализует SCADA КРУГ-2000?
2. Чем достигается открытость SCADA КРУГ-2000?
3. Какие механизмы SCADA КРУГ-2000 позволяют организовать обмен данными с различными устройствами, программами и системами?
4. Назовите основные структурные компоненты ПО SCADA КРУГ-2000.
5. Какие структурные компоненты ПО SCADA КРУГ-2000 относятся к среде разработки?
6. Какие структурные компоненты ПО SCADA КРУГ-2000 относятся к среде исполнения?
7. Какое дополнительное программное обеспечение используется в SCADA КРУГ-2000?
8. Из каких этапов состоит последовательность разработки АСУТП с помощью программных средств SCADA «КРУГ-2000»?
9. какое основное назначение Генератора базы данных системы Круг-2000?
10. Какие типы переменных реализованы в Генераторе базы данных системы Круг-2000?
11. Назначение Самописцев в Генераторе базы данных системы Круг-2000?
12. Какие типы самописцев поддерживаются в системе Круг-2000?
13. Каким образом происходит организация доступа к базе данных и системе в целом?
14. Какие типы печатных документов могут быть сформированы и выведены на печать в системе КРУГ-2000?
15. Для чего осуществляется верификация системы в Генераторе базы данных?
16. Для чего служит программа Генератор динамики системы КРУГ-2000?
17. Опишите алгоритм работы с Генератором динамики системы КРУГ-2000?
18. Назовите основные элементы интерфейса пользователя Генератора динамики.
19. Назовите основные типы объектов графической базы данных системы КРУГ-2000.
20. Для чего осуществляется верификация проекта в Генераторе динамики системы КРУГ-2000?

21. Какие объекты присутствуют в Генераторе динамики системы КРУГ-2000?
22. Назовите операнды технологического языка программирования КРУГОЛ системы КРУГ-2000.
23. Какие арифметические операции реализованы в технологическом языке программирования КРУГОЛ системы КРУГ-2000?
24. Какие логические операции реализованы в технологическом языке программирования КРУГОЛ системы КРУГ-2000?
25. Какие операции с таймерами реализованы в технологическом языке программирования КРУГОЛ системы КРУГ-2000?
26. Какие операторы реализованы в технологическом языке программирования КРУГОЛ системы КРУГ-2000?
27. Каким образом реализована процедура в технологическом языке программирования КРУГОЛ системы КРУГ-2000?
28. Каким образом реализована функция в технологическом языке программирования КРУГОЛ системы КРУГ-2000?
29. Из каких элементов состоит интегрированная среда разработки СТ технологического языка программирования КРУГОЛ системы КРУГ-2000?
30. Каков порядок разработки проекта в редакторе СТ?
31. Из каких элементов состоит интегрированная среда разработки ФБД технологического языка программирования КРУГОЛ системы КРУГ-2000?
32. Какие основные элементы языка ФБД?
33. Каким образом осуществляется настройка ОРС-клиента в системе КРУГ-2000?
34. Каким образом осуществляется привязка переменных БД системы КРУГ-2000 к переменным (тегам) контроллера?

ЗАКЛЮЧЕНИЕ

Прикладное программное обеспечение, используемое для автоматизации управления технологическими процессами, играет одну из главных ролей в повышении эффективности производства. Правильный выбор SCADA-системы во многом определяет дальнейший ход всего процесса автоматизации конкретного технологического процесса.

В учебном пособии рассмотрены одни из популярных коммерческих SCADA-систем российского производства: MasterSCADA (НПФ «ИнСАТ»), Круг-2000 (НПФ «Круг»).

Основная задача учебного пособия – дать студентам базовый уровень знаний, необходимых сегодня для разработки автоматизированных рабочих мест оператора, начиная от постановки задачи и заканчивая разработкой графического интерфейса и прикладных программ с использованием отечественных SCADA-систем.

СПИСОК ЛИТЕРАТУРЫ

1. **Технические** средства автоматизации. Программно-технические комплексы и контроллеры : учебное пособие / И. А. Елизаров, Ю. Ф. Мартемьянов, А. Г. Схиртладзе, С. В. Фролов. – М. : Изд-во Машиностроение, 2004. – 180 с.
2. **Системы** диспетчерского управления и сбора данных (SCADA-системы) // Мир компьютерной автоматизации. – 1999. – № 3. – С. 4 – 9.
3. **Андреев, Е. Б.** SCADA-системы: взгляд изнутри / Е. Б. Андреев, Н. А. Куцевич, О. В. Синенко. – М. : РТСофт, 2004. – 176 с.
4. **Пьявченко, Т. А.** Проектирование АСУТП в SCADA-системе : учебное пособие / Т. А. Пьявченко. – Таганрог : ТТИ ЮФУ, 2007. – 84 с.
5. **Куцевич, Н. А.** Компоненты и технологии в системах промышленной автоматизации / Н. А. Куцевич // Открытые системы. – 1999. – № 4. – С. 31 – 36.
6. **Прошин, Д. И.** Проблемы выбора инструментальных средств построения SCADA-систем / Д. И. Прошин, Л. В. Гурьянов // Информатизация и Системы Управления в Промышленности. – 2010. – № 1(25). – С. 21 – 25.
7. **Ефимов, И. П.** SCADA-система TraceMode / И. П. Ефимов, Д. А. Солуянов. – Ульяновск : УлГТУ, 2010. – 158 с.
8. **Trace Mode 6.** Быстрый старт. Руководство пользователя. [Электронный ресурс] – 2008. – URL : http://www.adastra.ru/files/documents/QUICK_START_v_6a_d.pdf.
9. **SCADA-система TRACE MODE.** [Электронный ресурс] – URL : <http://www.adastra.ru/products/dev/scada>
10. **Основные** преимущества MasterSCADA. [Электронный ресурс]. – URL : <http://insat.ru/products/?category=9>.
11. **Аблин, И. Е.** MasterSCADA: от простого к сложному / И. Е. Аблин // Промышленные измерения, контроль, автоматизация, диагностика. – 2007. – № 2. – С. 10 – 13.
12. Интерфейс инструментальной системы MasterSCADA [Электронный ресурс] – URL : <http://www.masterscada.ru/products/?category=1279>.
13. **Основы** проектирования в MasterSCADA : методическое пособие. – М. : Компания ИНСАТ, 2012. – 105 с.
14. **Редактор** мнемосхем MasterSCADA [Электронный ресурс]. – URL : <http://www.masterscada.ru/products/?category=1280>.

15. **Динамизация** в MasterSCADA. Обзор возможностей. – М. : Компания ИНСАТ, 2012. – 26 с.
16. **Тренды** MasterSCADA [Электронный ресурс] – URL : <http://www.masterscada.ru/products/?category=1281>.
17. **Редактор** отчётов Master SCADA [Электронный ресурс] – URL : <http://insat.ru/products/?category=1283>.
18. **SCADA** КРУГ-2000 – что это такое? [Электронный ресурс] – URL : <http://www.krug2000.ru/products/ppr/scada-2000/996.html>.
19. **Модульная** интегрированная SCADA КРУГ-2000™. Версия 3.0. ВВЕДЕНИЕ В КРУГ-2000. Руководство Пользователя. Редакция 3.
20. **Описание** модулей SCADA КРУГ-2000 [Электронный ресурс]. – URL : <http://www.krug2000.ru/products/ppr/scada-2000/1001.html>.
21. **Модульная** интегрированная SCADA КРУГ-2000™. Версия 4.1. БЫСТРЫЙ СТАРТ. Руководство Пользователя.
22. **Модульная** интегрированная SCADA КРУГ-2000™. Версия 4.1. ГЕНЕРАТОР БАЗЫ ДАННЫХ. Руководство Пользователя.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ОБЩИЕ ДАННЫЕ О SCADA-СИСТЕМАХ	4
1.1. Интегрированные системы управления производством	4
1.2. Состав SCADA-систем	6
1.3. Обзор SCADA-систем, представленных на рынке	10
1.4. Контрольные вопросы	19
2. MASTERSCADА	20
2.1. Общие сведения	20
2.2. Среда разработки MasterSCADA	25
2.2.1. Интерфейс среды разработки	25
2.2.2. Редактор мнемосхем MasterSCADA	32
2.2.3. Тренды MasterSCADA	36
2.2.4. Сообщения и журналы MasterSCADA	38
2.2.5. Редактор отчётов MasterSCADA	40
2.3. Среда исполнения MasterSCADA	42
2.4. Создание простейшего проекта	48
2.5. Контрольные вопросы	65
3. Модульная интегрированная SCADA КРУГ-2000	67
3.1. Общие сведения	67
3.2. Генератор базы данных системы КРУГ-2000	77
3.3. Генератор динамики системы КРУГ-2000	98
3.4. Технологический язык программирования системы КРУГ-2000	122
3.4.1. Основные правила и понятия технологического языка программирования – языка СТ	122
3.4.2. Среда разработки СТ	129
3.4.3. Среда разработки ФБД	131
3.4.4. Среда разработки ФБД	136
3.5. СОЗДАНИЕ ПРОСТЕЙШЕГО ПРОЕКТА	137
3.6. КОНТРОЛЬНЫЕ ВОПРОСЫ	154
ЗАКЛЮЧЕНИЕ	156
СПИСОК ЛИТЕРАТУРЫ	157

Учебное издание

ЕЛИЗАРОВ Игорь Александрович
ТРЕТЬЯКОВ Александр Александрович
ПЧЕЛИНЦЕВ Александр Николаевич
ПОГОНИН Василий Александрович
НАЗАРОВ Виктор Николаевич
ОНЕВСКИЙ Павел Михайлович

ИНТЕГРИРОВАННЫЕ СИСТЕМЫ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ: SCADA-СИСТЕМЫ

Учебное пособие

Редактор Е. С. Мордасова
Инженер по компьютерному макетированию Т. Ю. Зотова

ISBN 978-5-8265-1469-6



Подписано в печать 06.10.2015.
Формат 60×84 /16. 9,3 усл. печ. л.
Тираж 400 экз. (1-й з-д 50 экз.) Заказ № 435

Издательско-полиграфический центр
ФГБОУ ВПО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14
Тел. 8(4752) 63-81-08;
E-mail: izdatelstvo@admin.tstu.ru