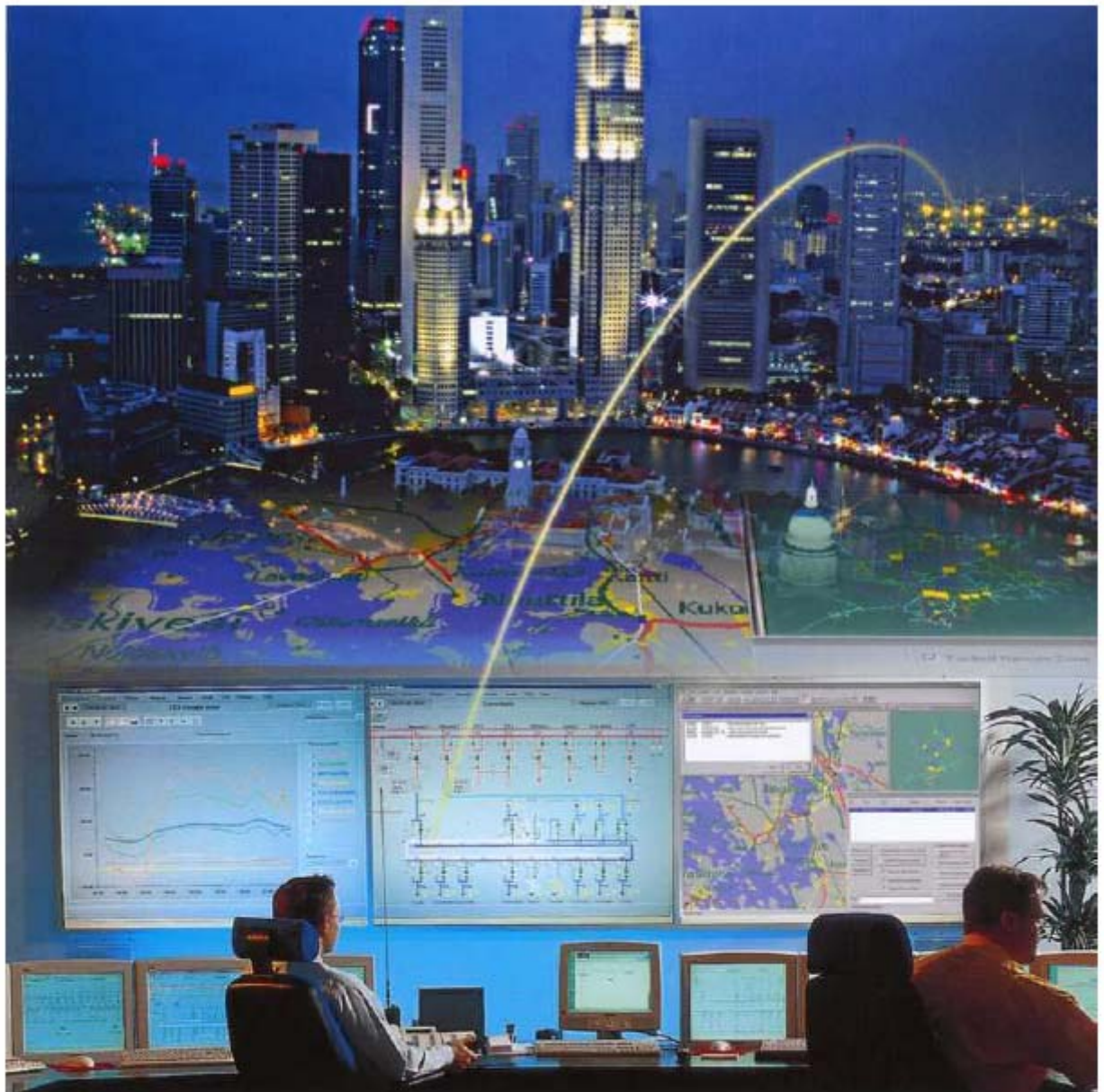


Система автоматизации и диспетчерского управления MicroSCADA



ABB

Мы оставляем за собой право изменить данные без предварительного уведомления.

Замечание 1

Информация, приведенная в этом документе, подлежит изменению без предварительного уведомления и не может быть истолкована в качестве обязательства со стороны АВВ. АВВ не несет ответственности за ошибки, которые могут встретиться в данном документе.

Замечание 2

Данный документ соответствует программной версии 8.4.4.

Замечание 3

Дополнительная информация, касающаяся последних исправлений, может быть найдена на носителях с программным обеспечением.

Торговые марки

Microsoft – зарегистрированная торговая марка Microsoft Corporation

Windows NT – торговая марка Microsoft Corporation

LonWorks - зарегистрированная торговая марка Echelon Corporation

Все права принадлежат их законным владельцам

Все перечисленные в данном документе программные продукты Microsoft являются зарегистрированными торговыми марками Microsoft Corporation

Документация по технологии MicroSCADA

Объекты приложения	1MRS7518848-MEN
Описание технологии MicroSCADA	1MRS751852-MUM
JAVA-API для MicroSCADA	1MRS751851-MEN
Язык программирования SCIL	1MRS751849-MEN
SCIL-API для MicroSCADA	1MRS752199-MEN
Коды состояния	1MRS751850-MEN
Конфигурирование системы	1MRS751846-MEN
Объекты системы	1MRS751847-MEN
MicroSCADA OPC DA клиент	1MRS752246-MEN

Документация по SYS 500

Установка SYS 500	1MRS751853-MEN
Редактор изображений	1MRS751854-MEN
Управление системой	1MRS751857-MEN
Объекты Visual SCIL	1MRS751856-MEN
Разработка интерфейса Visual SCIL	1MRS751855-MEN

Документация по COM 500

COM 500 Инжиниринг	1MRS751858-MEN
Подключение LONWORK устройств	1MRS751845-MEN
Прикладной интерфейс CPI	1MRS751859-MEN
Протокол DNP V3.00 Master	1MRS751860-MEN
Протокол DNP V3.00 Slave	1MRS751861-MEN
Протокол МЭК 60870-5-101 Master	1MRS751862-MEN
Протокол МЭК 60870-5-101 Slave	1MRS751863-MEN
Протокол МЭК 60870-5-103 Master	1MRS752012-MEN
Протокол МЭК 60870-5-104 Master	1MRS751964-MEN
Протокол МЭК 60870-5-104 Slave	1MRS751965-MEN
Протокол Modbus Master	1MRS752242-MEN
Протокол Modbus Slave	1MRS751864-MEN

Документация по LIB 500

Конфигурирование функций LIB 500	1MRS751880-MEN
Руководство оператора LIB 500	1MRS751885-MUM

Документация по LIB 510

Конфигурирование функций LIB 510	1MRS751886-MEN
Конфигурирование процесса CH LIB 510	1MRS751887-MEN
Руководство оператора LIB 500 (процесс CH)	1MRS751891-MUM
Руководство оператора LIB 510	1MRS751888-MUM

Документация по SMS 510

Установка SMS 510	1MRS751897-MEN
Руководство оператора SMS 510	1MRS751898-MUM

Документация по CAP 505

Установка CAP 505	1MRS751901-MEN
Руководство оператора CAP 505	1MRS751902-MUM
Инструмент конфигурации реле	1MRS751903-MEN
Редактор мнемосхем	1MRS751904-MEN
Конфигурация реле	1MRS751905-MEN
Инструмент конфигурации SPTO	1MRS751906-MEN
Редактор протоколов	1MRS751982-MUM

Общая документация по LIB, CAP, и SMS

Инструменты для терминалов и реле	1MRS752008-MUM
-----------------------------------	----------------

1	Введение	1
2	Технология MicroSCADA	2
3	Системный сервер SYS 500	3
4	Система связи	4
5	Прикладные библиотеки LIB 5xx	5

Содержание:

1. Введение	1
2. Технология MicroSCADA	3
2.1. Общая информация	3
2.2. Продукты технологии MicroSCADA	4
2.3. Описание системы MicroSCADA	6
2.3.1. Многоуровневая и одноуровневая система	6
2.3.2. Однопользовательская и многопользовательская система	7
2.3.3. Резервированная система	7
3. Системный сервер SYS 500	9
3.1. Общая информация	9
3.2. Базовая система	10
3.3. Приложения	11
3.4. Базы данных	12
3.4.1. База данных процесса	12
3.4.2. Ретроспективная база данных	14
3.4.3. База данных отчетов	14
3.5. Программирование на SCIL	15
3.5.1. Объекты	16
3.5.1.1. Объекты интерфейса пользователя	17
3.5.1.2. Объекты приложения	17
3.5.1.3. Системные объекты	18
3.5.2. Атрибуты	19
3.5.3. Visual SCIL	20
3.5.4. Объекты Visual SCIL	20
3.6. Графический интерфейс пользователя	21
3.6.1. Изображения	21
3.7. Основные инструменты	22
3.7.1. Инструменты конфигурации системы	23
3.7.2. Инструменты объектов приложения	24
3.7.3. Инструмент интерфейса пользователя	24
3.7.4. Вспомогательные инструменты	26
3.8. Периферийное оборудование	26
4. Система связи	29
4.1. Общая информация	29
4.1.1. Связь с верхним уровнем	30
4.1.2. Связь с процессом	30
4.1.3. Сетевая топология	31
4.2. Сервер связи COM 500	32
4.3. Выделенные серверы связи	35
4.4. Узел NET	36

4.5. Компоненты системы связи	37
4.5.1. Программное обеспечение	37
4.5.2. Сети	38
4.5.3. Аппаратное обеспечение	40
4.6. Устройства процесса	41
5. Прикладные библиотеки LIB 5xx	43
5.1. Разработка приложения	43
5.2. Прикладные библиотеки	44

1.**Введение****1**

Данное руководство является частью документации по технологии MicroSCADA 8.4.4.

Это руководство следует прочитать перед тем, как в первый раз приступать к работе с системой MicroSCADA. Данное руководство представляет собой введение в технологию MicroSCADA и служит основой для всей остальной документации.

Вся документация по технологии MicroSCADA 8.4.4 может быть разделена на следующие три категории:

- Руководства по технологии MicroSCADA. Эти документы предоставляют информацию общую для всех продуктов MicroSCADA
- Руководства по SYS 500. Эти документы предоставляют информацию, связанную с системным сервером SYS 500
- Руководства по COM 500. Эти документы предоставляют информацию, связанную с сервером связи COM 500

Использование данного руководства

Данный документ представляет собой введение в технологию MicroSCADA, ее компоненты и функции. В документе содержится общая информация о системе MicroSCADA, используемых технологиях, рабочих станциях, приложениях и графическом интерфейсе пользователя.

Разделы

Данный документ содержит следующие разделы:

- Введение, дающее сведения об использовании данного руководства
- Технология MicroSCADA – раздел, содержащий обзорную информацию о продуктах MicroSCADA и типах систем
- Системный сервер SYS 500 – раздел, описывающий SYS 500, его компоненты и функции
- Система связи – раздел, описывающий функциональность узлов NET, фронтэндов и серверов связи.
- Прикладные библиотеки LIB 5xx – раздел, описывающий процесс инжиниринга системы

2. Технология MicroSCADA

2

В современных условиях потребность в достоверной информации и надежном управлении процессом энергоснабжения постоянно возрастает. Ориентированная на соответствие этим требованиям, технология MicroSCADA предоставляет решение, которое ведет к увеличению результативности, надежности и эффективности вложения средств.

2.1. Общая информация

MicroSCADA – это платформонезависимая, программируемая, распределенная система управления и сбора данных (SCADA). Под платформонезависимостью понимается возможность использования в качестве аппаратной платформы для построения системы любых коммерчески доступных PC-компьютеров.

Гибкость системы достигается благодаря встроенному языку программирования высокого уровня SCIL (Supervisory Control Implementation Language), посредством которого создаются прикладные программы, и выполняется настройка системных функций. Более подробная информация о SCIL приведена в Разделе 3.5.1.

MicroSCADA может рассматриваться в качестве сети, в которой взаимодействие между управляющими узлами и распределенным процессом энергоснабжения происходит через модули системы связи.

Общая программная платформа, на которой базируется технология MicroSCADA, используется для создания как прикладных, так и системообразующих компонентов. SYS 500, COM500, LIB 510 – примеры продуктов технологии MicroSCADA.

В электроэнергетике MicroSCADA применяется для управления процессами передачи и распределения электроэнергии. Кроме того, MicroSCADA используется в таких областях деятельности, как теплоснабжение, водоподготовка и водоснабжение, очистка сточных вод, а также для управления нефте- и газопроводами.

Основная область применения системы MicroSCADA – автоматизация распределительных подстанций всех классов напряжения и электрической части станций. На базе технологии MicroSCADA возможно создание диспетчерских систем управления распределительными сетями различных уровней. Именно эти аспекты применения MicroSCADA нашли отражение в данном руководстве.

2.2.

Продукты технологии MicroSCADA

Иерархия продуктов

2

Иерархия продуктов технологии MicroSCADA показана на Рисунке 2.2.-1

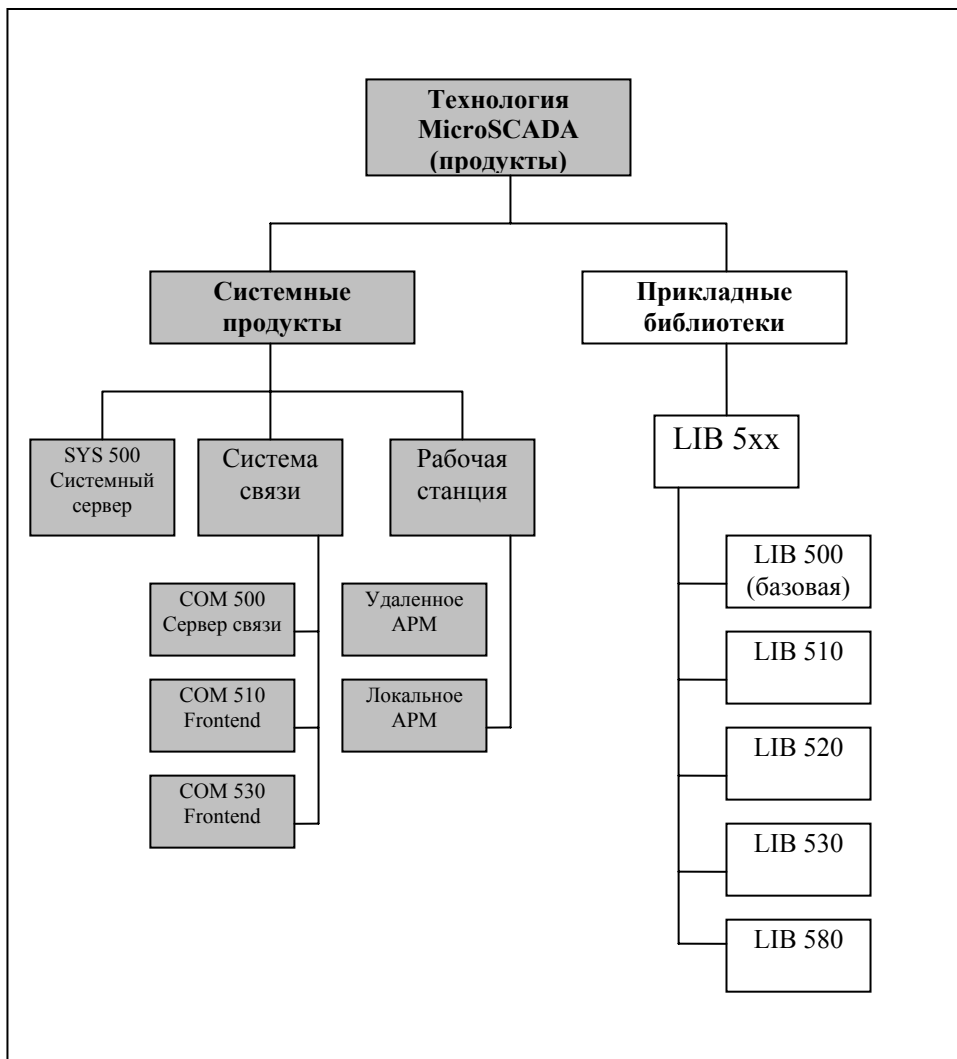


Рис. 2.2.-1 Продукты технологии MicroSCADA подразделяются на два семейства: системные продукты и прикладные библиотеки.

Продукты технологии MicroSCADA подразделяются на два семейства: системные продукты и прикладные библиотеки. Системные продукты служат стержневыми компонентами системы. Прикладные библиотеки сгруппированы в LIB пакеты (описаны в Разделе 5). Продукты из различных семейств могут использоваться совместно. Некоторые из них требуют наличия других продуктов MicroSCADA, другие являются самодостаточными (SYS 500 и COM 500).

Системные продукты

Системные продукты – системный сервер SYS 500, система связи и рабочие станции операторов – включают в себя необходимое аппаратное и программное обеспечение для запуска приложений и обработки связи между компонентами системы. SYS 500 описан в Разделе 3, система связи описана в Разделе 4.

Рабочая станция оператора

Рабочая станция оператора – это компьютер, на котором располагается автоматизированное рабочее место (АРМ). АРМ используется для контроля и управления первичным процессом (посредством изображений и диалогов). Рабочее место может быть интегрировано с системным сервером SYS 500 либо располагаться на отдельном компьютере, подключенном к серверу. Рабочие станции могут подключаться через удаленный доступ по коммутируемым линиям либо по локальной сети (LAN).



Компьютер, выступающий в качестве рабочей станции, может одновременно использоваться и для других целей.

Рабочая станция оператора состоит из компьютера, операционной системы, программного обеспечения MicroSCADA Workstation и принтера. Взаимодействие оператора с системой производится через монитор, клавиатуру и «мышь».

Программное обеспечение АРМ

Программное обеспечение АРМ (MicroSCADA Workstation Software) – это набор программ, используемых для контроля и управления процессом. Оно позволяет принимать системные сообщения и запускать Мониторы MicroSCADA.

Мониторы MicroSCADA

Монитор MicroSCADA – это программный интерфейс для отображения приложения MicroSCADA на экране компьютера. Он необходим для взаимодействия оператора с базовым сервером системы. Мониторы MicroSCADA бывают двух типов – Visual SCIL-мониторы и X-мониторы. MicroSCADA мониторы подключаются к SYS 500. Изображения и диалоги, выводимые на экран рабочей станции, всегда сохраняются в системном сервере.

Базовые функции системного сервера SYS 500 (контроль процесса и регистрация данных) могут выполняться в фоновом режиме и не требуют открытия мониторов.

2.3.2.**Однопользовательская и многопользовательская система****2**

Однопользовательская и многопользовательская системы строятся на базе SYS 500. Однопользовательская система имеет один локальный монитор MicroSCADA и узел NET. Для связи с низовыми устройствами может использоваться модем. В многопользовательской системе одна либо несколько рабочих станций подключаются к SYS 500 (обычно посредством локальной сети). К системе также может быть подключен выделенный коммуникационный сервер (frontend), используемый для увеличения количества линий связи.

Узлы NET применяются в однопользовательских системах для увеличения числа доступных протоколов и COM-портов. Кроме того, они уменьшают нагрузку на базовый компьютер системы.

Более детальная информация о SYS 500 приведена в Разделе 3.

2.3.3.**Резервированная система**

Одиночная система MicroSCADA включает в себя лишь по одному экземпляру каждого системного компонента. Резервированная система содержит две базовые системы и/или два узла NET и/или два соединения с локальной сетью. Главная идея резервирования – сделать систему более надежной путем дублирования основных модулей. В большинстве случаев, важнейшим требованием к надежности является необходимость полного сохранения функциональности системы при выходе из строя отдельных компонентов. Это означает, что в резервированной системе в случае повреждения какого-либо компонента основной базовой системы, резервная система распознает аварийную ситуацию и немедленно переводит управление «на себя».

3. Системный сервер SYS 500

В этом разделе

В этом разделе приведена вводная информация о SYS 500 и его компонентах.

3

3.1. Общая информация

Системный сервер SYS 500 является ядром программируемой автоматизированной системы на базе PC. Компьютер, где установлен системный сервер, называется компьютером SYS 500. Этот компьютер может рассматриваться как главный компьютер. SYS 500 может быть оснащен интерфейсом человек-система (Human-System Interface - HSI) либо может использоваться без него (Рисунок. 3.1.-1).

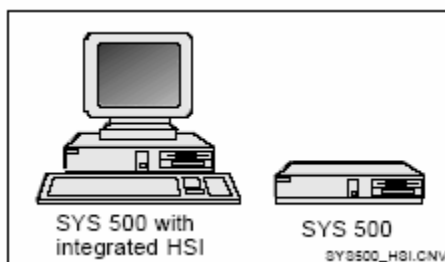


Рис. 3.1.-1 SYS 500 с интерфейсом HSI и без HSI

Задачи

Системный сервер SYS 500 запускает операционную систему Windows NT/2000 и программное обеспечение MicroSCADA. SYS 500 состоит из аппаратной и программной частей, необходимых для взаимодействия между различными узлами системы. Компьютер, на который возложены функции системного сервера, является главным компьютером. Основные функции главного компьютера – сбор данных о процессе, управление и самодиагностика системы.

Функции

Системный сервер реализует следующие функции:

- Контроль и управление первичным процессом (посредством графического интерфейса пользователя, автоматическое циклическое и реакционно-событийное)
- Обработку событий и тревог
- Генерацию отчетов по результатам расчетов и измерений
- Доступ к данным и настройкам в соответствии с правами пользователя
- Поддержку многозадачности приложений

- Документирование и оперативный ввод данных
- Поддержку многопользовательского режима
- Печать
- Обмен данными с другими приложениями

3

Компоненты

Программное обеспечение системного сервера состоит из:

- Операционной системы Windows NT/2000 (Server, Professional или Workstation)
- Базовой системы
- Базовых инструментов
- Приложения
- ПО Hummingbird eXceed (если используются удаленные АРМ)

Система связи и рабочая станция оператора могут быть интегрированы в SYS 500 (Рисунок 3.1.-2).

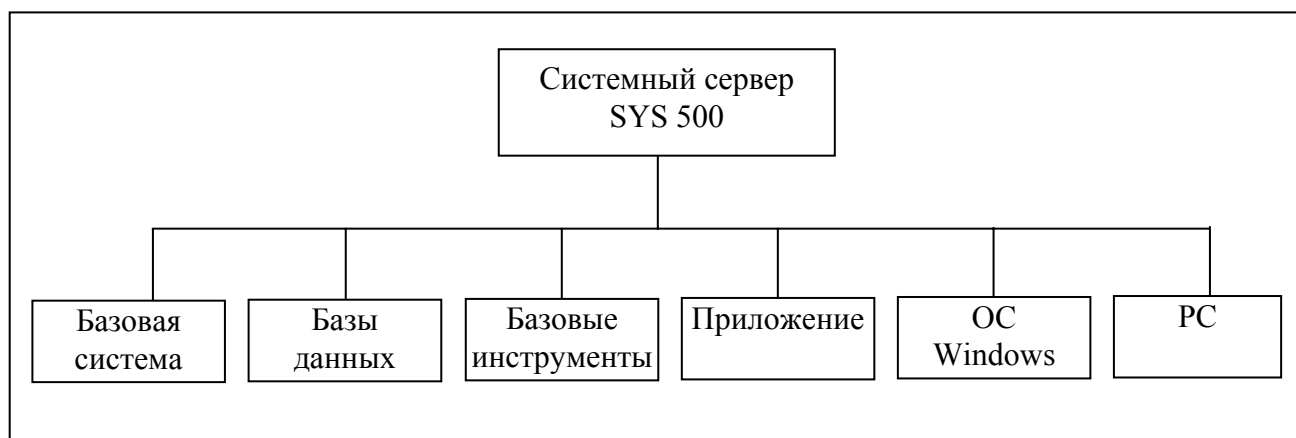


Рис. 3.1.-2 Основные компоненты системного сервера SYS 500

3.2.

Базовая система

Базовая система собирает данные, характеризующие протекание контролируемого процесса, посредством узлов связи NET и сохраняет их в базе данных процесса. Собранная информация используется для визуализации через HSI на экранах рабочих станций, архивирования, расчетов, печати и для последующей обработки и передачи в другие системы. Соответствующим путем, управляющие команды передаются из базы данных процесса через устройства нижнего уровня на исполнительные механизмы. Команды могут быть получены от операторов системы через HSI, сгенерированы автоматическими функциями или другими АСУ. База данных процесса всегда отражает текущее состояние параметров контролируемого процесса.

Базовая система формирует окружение для запуска приложения (Рис. 3.2.-1). Приложение реализует необходимую функциональность, в то время как базовая система лишь предоставляет некоторый набор стандартных сервисов. Доступ к ним может быть получен посредством SCIL. Более подробная информация о SCIL приведена в Разделе 3.5.1.

3

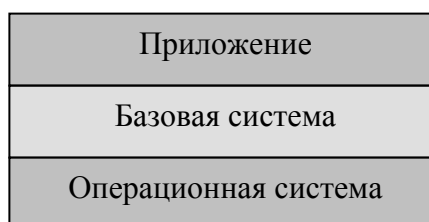


Рис. 3.2.-1 Уровни ПО SYS 500. (Приложение выполняется базовой системой.)

Базовая система также представляет программный интерфейс (API) для подключения дополнительных функций, реализованных как отдельные программные процедуры.

3.3.

Приложения

Структура и вид приложения определяется требованиями заказчика и заданным алгоритмом управления первичным процессом. Это означает, что именно приложение воплощает конкретный интерфейс пользователя и реализует специфическую функциональность системного сервера SYS 500.

Программное обеспечение приложения состоит из:

- Изображений и диалогов
- Базы данных процесса, ретроспективной базы данных и базы данных отчетов
- Управляющих программ
- Файлов конфигурации

Специфические прикладные функции реализуются в процессе инжиниринга приложения. Приложение может быть создано при помощи стандартной прикладной библиотеки LIB 500 либо путем программирования на SCIL.

Все функции приложения сосредоточены на его программном уровне. Изменения, вносимые в приложения, не затрагивают уровень базовой системы. Таким образом, базовая система может исполнять несколько приложений одновременно. Каждое приложение может контролировать свой первичный процесс по своим независимым каналам связи или разделять их с другими приложениями.

Приложения могут обмениваться данными между собой.

3.4. Базы данных

Каждое приложение содержит базу данных процесса, базу данных отчетов и ретроспективную базу данных. База данных процесса используется для контроля над первичным процессом, база данных отчетов для хранения расчетных данных и автоматической активации программ. Ретроспективная база данных является основой функции списка событий. Эти три базы данных более подробно описываются в следующих разделах.

3

3.5. База данных процесса

База данных процесса содержит объекты приложения. В объектах приложения регистрируются значения входных и выходных сигналов процесса. Обновление объектов происходит в реальном масштабе времени. Связь первичного процесса с базовой системой осуществляется через объекты процесса. Схема информационного потока показана на Рисунке. 3.4.1.-1.

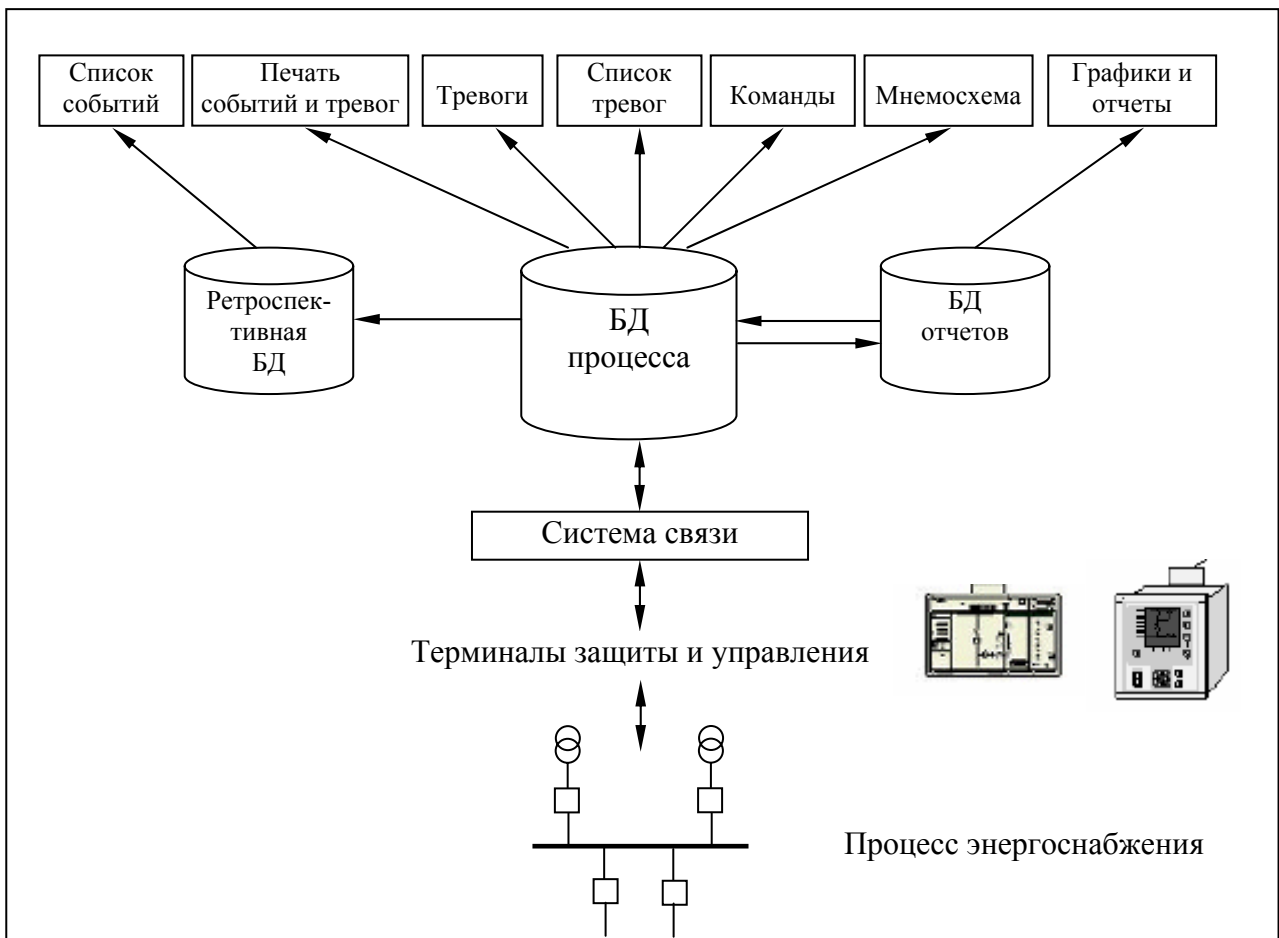


Рис.3.4.1.-1 Информационный поток от процесса к оператору и обратно проходит через базу данных процесса.

События

Событие информирует оператора о том, что что-то произошло в системе. Примерами событий являются изменения значений объектов, тревоги и предупреждения, описание тревог. События могут вызывать печать данных, активизацию функций автоматического управления, запись в список событий и ретроспективную БД.

3

Список событий содержит все произошедшие в системе события. Кроме того, он информирует о действиях других пользователей, манипуляциях с объектами, квитировании тревог, редактировании пределов измерений и т.п. При помощи LIB 500 пользователь может оперативно создавать собственные фильтры для отбора необходимой информации из списка событий. Фильтр может строиться по одному или нескольким критериям поиска данных.

Тревоги

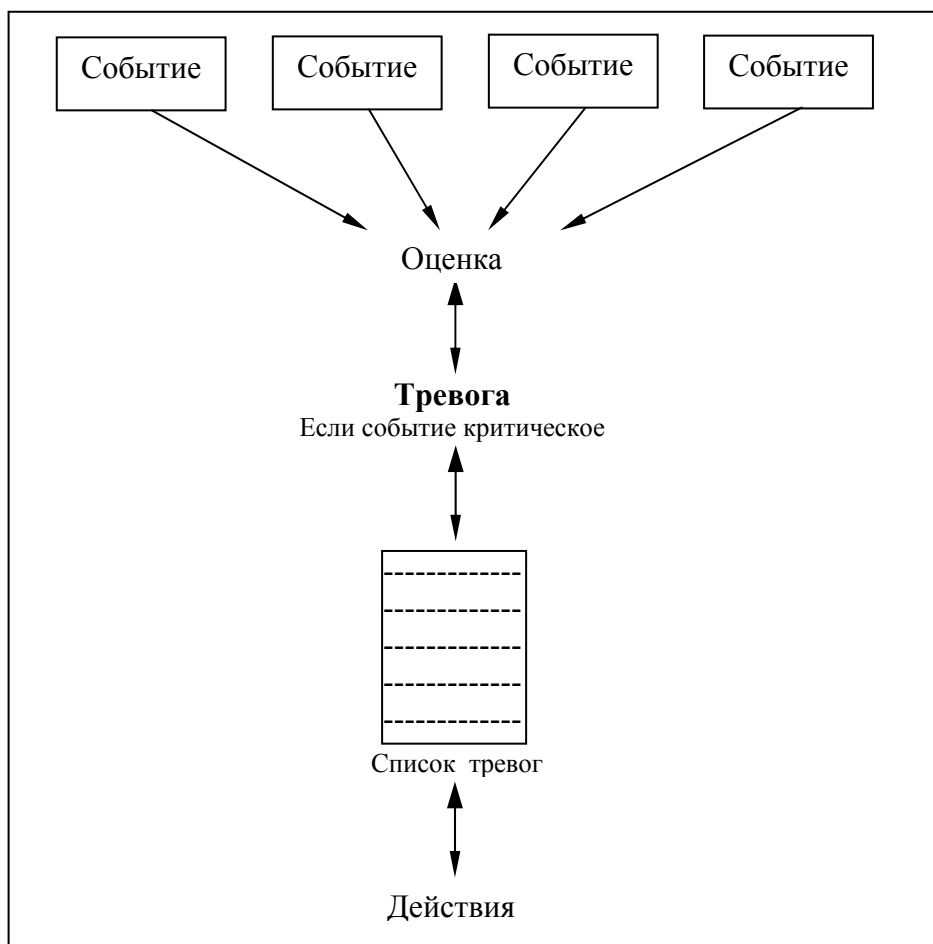
Тревоги генерируются в случае возникновения в системе опасных ситуаций. Тревоги являются подмножеством событий. Тревоги могут активизировать аудио-видео сигнализацию, изменения в мнемосхеме процесса, регистрацию в списке тревог, выводить аварийные изображения на экран и данные на печать. Информация об аварийных объектах помещается в буфер тревог. Объект находится в буфере, пока причина тревоги не исчезнет либо тревога не будет квитирована.

Список тревог отображает на экране рабочей станции информацию из буфера тревог. Список тревог разделяется на две части: активные и прошедшие (fleeting) тревоги. Сигнал тревоги всегда снабжается описательным текстом, объясняющим причину тревоги. Схема обработки тревог показана на Рисунке 3.4.1.-2.

Тревоги и события могут генерироваться тремя способами:

- События в контролируемом процессе могут генерировать тревоги. Параметры процесса оцениваются базовой системой согласно установленных пределов. Например, если измеренное значение выходит из заданного диапазона, возникает тревожный сигнал.
- Сама система может генерировать внутренние тревоги из диагностических программ, которые контролируют состояние компонентов MicroSCADA.
- Системная тревога может быть сгенерирована внешним модулем, выполняющим сторожевую функцию. Тревоги этих типов не включаются в список тревог.

Другие устройства в системе MicroSCADA также могут служить источниками тревог.



3

Рис. 3.4.1.-2 Схема обработки аварийных ситуаций. Тревоги анализируются и генерируются в базе данных процесса

3.4.2.

Ретроспективная база данных

События сохраняются в ретроспективной базе данных как история событий. Ретроспективная БД состоит из файлов, каждый из которых хранит события за одни сутки. Эта база данных служит основой функции списка событий библиотеки LIB 500 (ревизия 4.0.2. и новее). Каждое событие в ретроспективной базе данных содержит большинство значений атрибутов соответствующего объекта процесса и некоторые специфические атрибуты.

3.4.3.

База данных отчетов

База данных отчетов хранит архивные данные и рассчитанные значения. В этой БД сохраняются SCIL-программы, активизируемые каналами событий и каналами времени, а также сами эти каналы и объекты данных.

Данные для отчетов в БД отчетов должны быть обработаны математически либо статистически. Собранные из разных источников, данные подвергаются анализу и расчету, для записи алгоритма обработки применяются SCIL-выражения. Затем данные записываются в базу данных. По ним составляются оперативные отчеты, отчеты по тревогам и событиям, тренды и т.д.

3

Рисунок 3.4.3.-1 иллюстрирует алгоритм обработки отчетных данных.

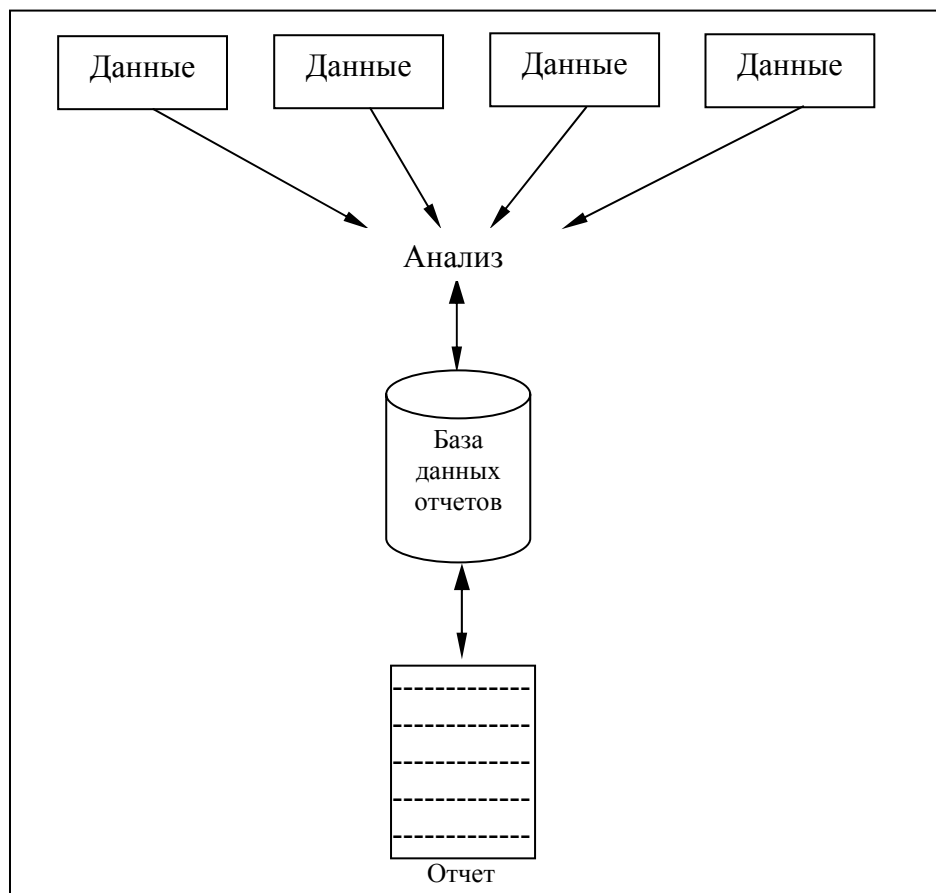


Рис. 3.4.3.-1 Поток данных из различных источников в отчеты

3.5.

Программирование на SCIL

Программирование в MicroSCADA осуществляется на SCIL – высокоуровневом прикладном языке программирования. SCIL-программы, создаваемые в SCIL-редакторе, содержат инструкции, исполняемые системой. Программы можно связать с изображениями, командными процедурами, Visual SCIL-диалогами и их элементами. Поскольку MicroSCADA построена на SCIL, SCIL-программы присутствуют в каждом приложении MicroSCADA. Следует отметить, что SCIL – оригинальный язык и может использоваться только в среде MicroSCADA.

Как и в большинстве языков программирования, в SCIL используются такие конструкции как присваивание переменных, арифметические выражения, сравнения, вложенные блоки, циклы, выбор по условию и т.д. SCIL является интерпретируемым языком, тем не менее SCIL-компилятор также присутствует.

SCIL-директивы состоят из:

- Команд (!SHOW, #SET, #ON и т.д.)
- Объектов (например, объекта процесса)
- Выражений (TIMES, SECOND, CLOCK и т.д.)
- Переменных (@init, %use)
- Имен (изображений и диалогов)

Существует три типа SCIL-команд. Команды изображений всегда начинаются с “!”, полнографические и motif-команды с “.”, а команды управляющие исполнением программы с “#”.

Ошибки синтаксиса проверяются редактором автоматически, во время написания программы, что избавляет от необходимости последующего синтаксического тестирования дополнительными инструментами. Синтаксические ошибки идентифицируются уникальными кодами, их расшифровка может быть найдена в руководстве Status Codes (1MRS751850-MEN). Далее приведен пример SCIL-программы:

```
!show the trend basket dialog
!WIN_NAME FORMAT/BASKET
!WIN_BG_COLOR FORMAT/BASKET ("M",1)
!WIN_POS FORMAT/BASKET (10,15)
!WIN_PIC FORMAT/BASKET F_USE/FTU_BASKB
!SHOW FORMAT/BASKET
```

Предопределенные функции

SCIL содержит большое количество готовых функций для обработки различных типов данных. Например, программисту доступны следующие категории:

- Арифметические функции
- Функции времени
- Функции оперирования базами данных
- Файловые функции

3.5.1.

Объекты

MicroSCADA – это объектно-ориентированная среда. Под объектом понимается некоторая программируемая сущность, воплощающая что-нибудь. В MicroSCADA объекты представляют модули процесса, системные функции или SCIL-программы. Объект определяется своими атрибутами. Следующие три типа объектов – объекты

интерфейса пользователя, объекты приложения и системные объекты – описываются в этом разделе.

3.5.1.1.

Объекты интерфейса пользователя

Интерфейс пользователя в системе MicroSCADA реализован в мониторе MicroSCADA. Объекты интерфейса пользователя формируют интерфейс приложения. К этим объектам относятся изображения, Visual SCIL-диалоги и элементы диалогов. Все они программируются и управляются посредством SCIL. Объекты интерфейса группируются в изображения и Visual SCIL объекты. Интерфейс пользователя более детально описан в Разделе 3.6.

3

3.5.1.2.

Объекты приложения

Объекты приложения – это программируемые элементы, которые выполняют различные задачи, такие как контроль первичного процесса в реальном масштабе времени, управление, регистрацию и хранение данных, расчеты и автоматическую активацию программ по времени и событиям.

Объекты приложения подразделяются на следующие типы:

- **Шкалы** Содержат алгоритмы для масштабирования измеренных значений, получаемых от устройств нижнего уровня. Могут использоваться эквивалентные, линейные и интерполяционные шкалы. Несколько объектов процесса могут использовать одну шкалу.
- **Объекты данных** Регистрируют и хранят выбранные и рассчитанные данные. Каждое зарегистрированное значение снабжается меткой времени и кодом состояния, которые описывают достоверность значения. Объекты данных используются для построения графиков, генерации отчетов по измерениям, хранения данных системной конфигурации, расчетов, анализа и т.д.
- **Командные процедуры** Содержат SCIL-программы, которые могут выполняться вручную и автоматически. Применяются для любых типов автоматических операций – расчетов, управления первичным оборудованием, печати отчетов, конфигурации системы и т.д. Могут автоматически запускаться при старте системе.
- **Каналы времени** Каналы времени используются для автоматического запуска программ и регистрации данных. Активация происходит в определенные моменты времени (например, каждый час). Один канал времени может выполнять несколько действий в соответствии с их приоритетом. Активация может выполняться циклически либо по абсолютным значениям (например, в 17:35 определенного числа текущего года).
- **Каналы событий** Каналы используются для автоматического запуска программ, каналов времени и регистрации данных. Обычно, каналы событий активизируются событиями процесса, отражающимися в изменении значений атрибутов объектов процесса. Каналы событий передают информацию о событиях процесса из базы

- данных процесса в базу данных отчетов, где они вызывают определенные действия.
- **Объекты событий** Объекты событий вызывают автоматическое обновление в изображениях. Объекты событий могут быть активизированы вручную либо автоматически. Атрибутов не имеют.
 - **Временные объекты** Представляют собой списки для временного хранения атрибутов и значений атрибутов, используемых в SCIL-программах
 - **Объекты процесса** В объектах процесса отражается реальное состояние модулей процесса – выключателей, разъединителей, переключателей, терминалов РЗА, датчиков, преобразователей и контроллеров.
- В общем случае, каждый входной и выходной сигнал представляется отдельным объектом процесса в БД процесса. Объекты, относящиеся к одному модулю или присоединению, объединяются в группу. Группа содержит индексированные объекты с одним логическим именем.
- Большая часть функциональности базы данных процесса сосредоточена в объектах процесса. Обработка входной информации и выдача управляющих воздействий осуществляется через объекты процесса. Кроме самого значения сигнала, в объекте процесса сохраняется вся информация, связанная с этим сигналом. Доступ к ней осуществляется посредством атрибутов объекта.

3.5.1.3.

Системные объекты

Системные объекты – это программируемые элементы, описывающие конфигурацию и связи компонентов системы MicroSCADA. Существует два типа системных объектов: объекты базовой системы и объекты системы связи. Эти объекты описывают конфигурацию системы совместно с данными конфигурации PC-NET. Доступ к объектам осуществляется через их атрибуты из SCIL.

Объекты базовой системы предназначены для доступа к настройкам базовой системы. Они определяют логические связи с узлами NET, другими базовыми системами и приложениями. При помощи системных объектов можно изменять конфигурацию системы и логические связи с другими устройствами. Каждая базовая система имеет свой набор системных объектов.

Объекты системы связи и их атрибуты описывают конфигурацию узлов NET. Кроме того, они предоставляют узлам NET описание линий связи и подключенных к ним устройств. Объекты системы связи предназначены для организации связи с первичным процессом.

3.5.2.

Атрибуты

Атрибут описывает свойство объекта и связанную с ним информацию. В атрибутах сохраняются значение объекта, его функции, свойства и активность. Обычно, объект обладает довольно большим количеством различных атрибутов. Каждый атрибут доступен из SCIL. Наборы атрибутов специфичны для каждого объекта.

3

Статические и динамические атрибуты

Атрибуты можно разделить на конфигурируемые и динамические. Смотреть Рисунок 3.5.2.-1.

- Конфигурируемые атрибуты идентифицируют объект и определяют его свойства. Примерами таких атрибутов являются имя объекта, его адрес, критерий активизации, связь с другими объектами и обработка тревог. Значения конфигурируемых атрибутов можно установить из SCIL либо при помощи инструмента описания объектов.
- Динамические атрибуты отображают значение объекта и некоторую дополнительную информацию, связанную с ним – код состояния и метку времени. Динамические атрибуты обычно устанавливаются системой по факту события в процессе.

ОБЪЕКТ			
Конфигурируемые атрибуты		Динамические атрибуты	
Name	SAMPLE	Data	0,785
Copied from	ORIGIN	Validation stamp	OK
Activated	TC 1h	Time stamp	12:23:34:347

Рис. 3.5.2.-1 Объект с конфигурируемыми и динамическими атрибутами

Атрибуты объекта процесса

Управление первичным процессом осуществляется чрез изменение значений объектов процесса.

Обновленное значение объекта процесса сохраняется вместе с дополнительной информацией. Например, сохраняется время последнего обновления. Обновление объекта может быть выполнено из SCIL. Обновление атрибута объекта процесса может привести к последующим действиям:

- Генерации тревог
- Выводу на печать
- Обновлению изображений
- Активизации канала событий
- Активизации объекта событий
- Регистрации данных

3.5.3. Visual SCIL

Visual SCIL – это объектно-ориентированная среда программирования в стиле Windows. Диалоги Visual SCIL программируются через методы. Методы, за исключением некоторых predefined, представляют собой SCIL-программы.

Методы определяют динамические операции в диалогах и элементах диалогов. Каждый диалог и элемент диалога имеет свой набор методов. Методы могут выполняться циклически либо в качестве реакции на определенные события или действия пользователя. Некоторые методы можно вызвать из SCIL.

Редактируемые методы программируются в SCIL-редакторе, доступном из Редактора Диалогов.

3.5.4. Объекты Visual SCIL

Существует около 40 типов объектов Visual SCIL, которые могут быть использованы для построения графического интерфейса пользователя.

Типы объектов Visual SCIL

Объекты Visual SCIL можно сгруппировать по следующим основным типам:

- Объекты-контейнеры. Объекты этой группы могут содержать в себе другие элементы диалогов. К этой группе относятся контейнеры, контейнеры изображений, меню, закладки и страницы закладок.
- Элементы диалогов (например, кнопка)
- Рисунки

Кнопки, текстовые поля, меню и рисунки называются элементами диалога.

Создание объектов Visual SCIL

Диалоги обычно создаются и программируются в Редакторе Диалогов, используя широкий набор стандартных элементов: надписей, текстовых полей, кнопок, списков, полос прокрутки, панелей меню.

Объекты Visual SCIL могут загружаться из файлов либо создаваться программным путем из диалогов и изображений.

Система диалогов

Объекты Visual SCIL, например диалоги, используются в основном для создания инструментов конфигурирования системы. Объекты показываются в виде иерархической структуры, где корневым объектом является главный диалог или контейнер изображения. Главный диалог и все остальные вложенные диалоги, открытые из него либо из элементов изображения, принадлежат к одной системе диалогов.

3

3.6.

Графический интерфейс пользователя

Графический интерфейс пользователя системного сервера SYS 500 складывается из изображений и диалогов. Изображения – это динамические рисунки, состоящие из статического фона, динамических функций, окон и функциональных ключей. Изображения применяются для управления и отображения мнемосхемы процесса.

Диалоги, созданные в Visual SCIL, используются главным образом для создания инструментов прикладного инжиниринга системы.

3.6.1.

Изображения

Оператор использует изображения для контроля и управления процессом. Процесс визуализируется как однолинейная схема с различными символами и цветами.

Компоненты изображений

Изображение складывается из компонентов изображения – фона, окон, функциональных ключей, программ и функций изображения. Фон изображения содержит статические, неизменяемые графические элементы.

Окна – это динамические части изображения, которые могут быть показаны, удалены либо перемещены по экрану без влияния на другие компоненты изображения. Окна могут содержать вложенные изображения, текст и данные, рисунки и графики. Окна могут располагаться в любом месте изображения и перекрывать друг друга.

Функциональные ключи – программируемые прямоугольные области изображения. При щелчке мышью на функциональном ключе запускается прикладная программа либо открывается поле ввода данных. Программы определяют поведение изображений.

Когда стандартная функция устанавливается из библиотеки, ее копия вставляется в изображение как функция изображения.

Функция изображения – это набор компонентов изображения, который может включать в себя все упомянутые выше компоненты. Компоненты изображения показаны на Рисунке 3.6.1.-1. Изображение может содержать окна, которые также могут содержать изображения. Кроме того, в них могут содержаться функции изображения, сконструированные как изображения.

3

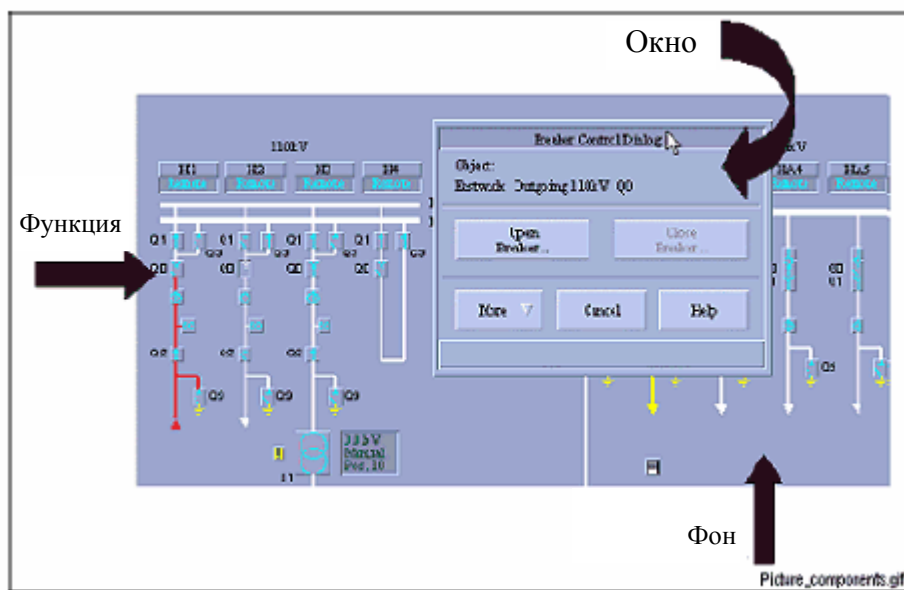


Рис. 3.6.1.-1 Компоненты изображения (программы не показаны)

3.7.

Основные инструменты

Базовые инструменты упрощают использование системы MicroSCADA. Например, конфигурацию системы можно выполнить из SCIL-программы, однако графические инструменты облегчают эту процедуру.

Инструменты используются для создания, модификации и просмотра компонентов системы и приложений. Базовые инструменты применяются для следующих задач:

- Редактирования изображений и диалогов
- Программирования на SCIL
- Управления объектами приложения
- Тестирования функциональности приложения
- Создания и модификации системных объектов
- Поиска текстовых строк в программах, изображениях и текстовых файлах

Инструменты могут быть сгруппированы по следующим категориям:

- Инструменты интерфейса пользователя
- Инструменты приложения
- Системные инструменты
- Вспомогательные инструменты

Эти группы детально описываются в следующих разделах. Инструменты запускаются из Tool Manager (см. Рисунок 3.7.1), где они размещаются на различных страницах.

Страницы инструментов могут быть модифицированы. Можно изменять названия страниц, добавлять либо удалять новые, перемещать инструменты между страницами, вставлять новые инструменты и т.д.

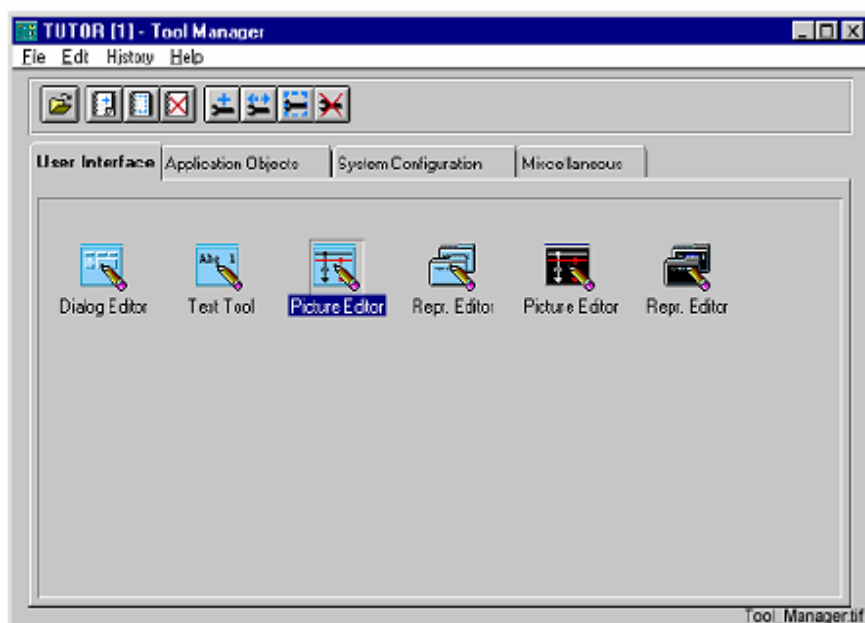


Рис. 3.7.-1 Tool Manager группирует значки для запуска различных инструментов.

3.7.1.

Инструменты конфигурации системы

Инструменты конфигурации системы используются для описания системных объектов, их атрибутов и связанных с ними функций. В эту группу входят инструменты Font Setting, Color Setting, Date/Time Settings, Base System Configuration и System Configuration.

3.7.2.

Инструменты объектов приложения

Инструменты приложения используются для определения объектов, атрибутов и других характеристик, которые тесно связаны с приложением и реализуют его функциональность. На странице Application Objects расположены Object Navigator и Doc./Export Tool.

3

Навигатор объектов (Object Navigator) позволяет просматривать списки объектов приложения, копировать, создавать и удалять различные типы объектов. Из навигатора объектов запускается Process Object Definition Tool (Рисунок 3.7.2.-1), который используется для создания и модификации объектов определенных типов.

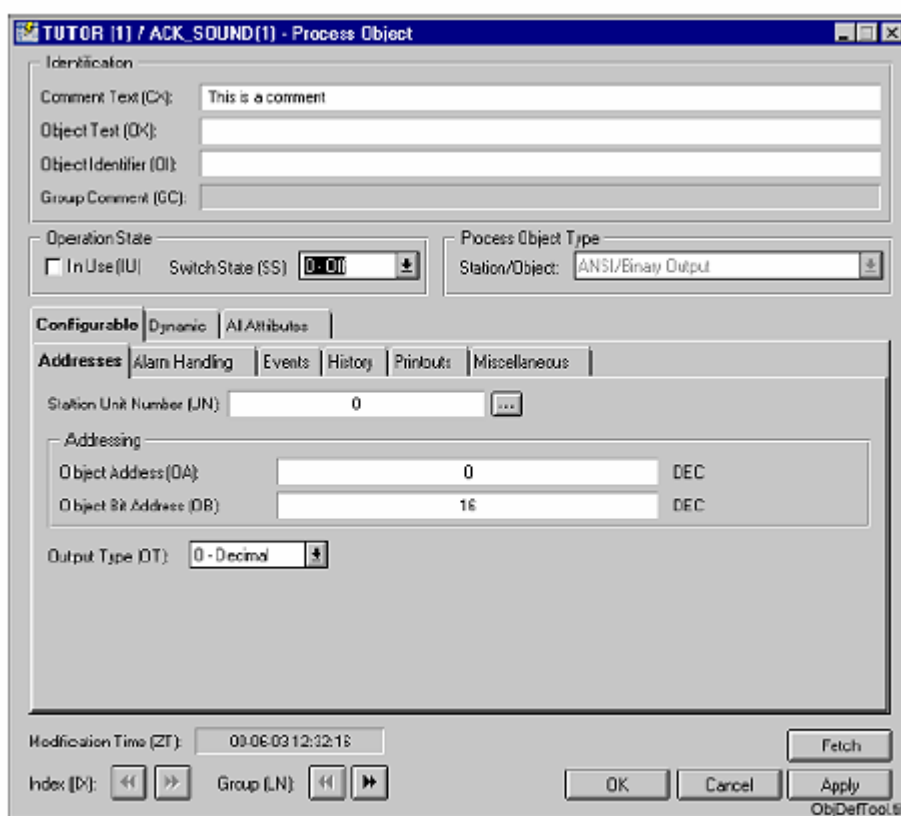


Рис. 3.7.2.-1 Инструмент описания объектов процесса (Process Object Definition Tool), запускаемый из навигатора объектов

3.7.3.

Инструменты интерфейса пользователя

Объекты интерфейса пользователя в основном используются для создания графического интерфейса приложения или других инструментов. В эту группу входят: редактор диалогов, редактор изображений (полнографический и псевдографический), инструмент перевода текстовых сообщений и редактор библиотеки представлений (полнографический и псевдографический).

Редактор изображений применяется для создания и программирования изображений. Редактирование осуществляется в четырех режимах: фоновая графика, функции изображения, функциональные ключи и окна. Режим редактирования зависит от выбранного компонента изображения. Например, редактор окон может быть запущен, если выделено како-то окно. Изображения создаются в самом начале процесса инжиниринга приложения, когда инженер проекта разрабатывает графику и реализует основную функциональность системы. Некоторые общие для всех приложений функции могут быть установлены из прикладных библиотек. Редактор изображений показан на Рисунке 3.7.2.-2.

Следует отметить, что между изображениями и рисунками в MicroSCADA существует определенная разница. Изображения – это динамические иллюстрации, содержащие кнопки, окна и программы, тогда как рисунки всегда статичны и могут включаться в изображения в качестве фона.

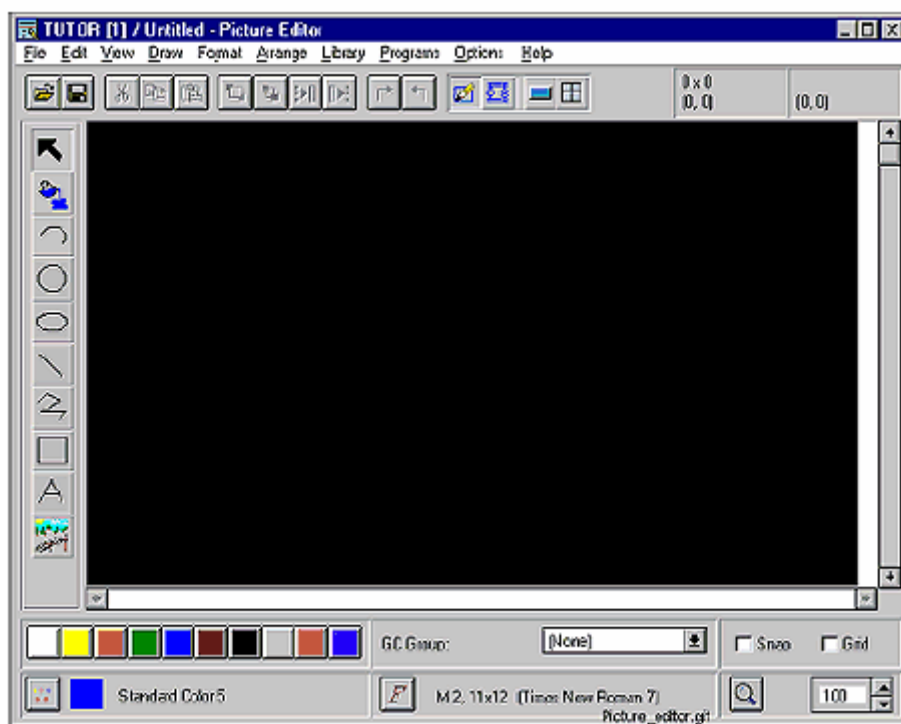


Рис. 3.7.2.-2 Редактор Изображений

Инструмент Representation Library Manager используется для копирования и переноса представлений функций изображений из одной библиотеки в другую. При помощи этого инструмента редактируются свойства функций изображения, которые нельзя изменить в редакторе изображений.

Редактор диалогов (Dialog Editor) применяется для создания и редактирования Visual SCIL диалогов.

Text Translation Tool позволяет переводить текстовые строки в

инструментах и диалогах (либо в изображениях LIB 5xx) на локальные языки.

3.7.4.

Вспомогательные инструменты

Вспомогательные инструменты, главным образом используемые для инжиниринга системы, собраны на странице Miscellaneous. К этой группе относятся редактор SCIL (SCIL Program Editor), Test Dialog, DB<-->Text Tool и Search Tool.

3

SCIL Program Editor – вспомогательный инструмент, специально разработанный для редактирования SCIL-программ. Он запускается из различных инструментов в том случае, если SCIL-программы должны быть отредактированы. Редактор SCIL можно открыть из Tool Manager для создания и редактирования текстовых файлов. SCIL Program Editor оснащен различными стандартными для среды разработки функциями – вставкой и копированием строк, текстовым поиском и заменой, инструментом просмотра дерева объектов и примеров вызова процедур и т.д. Программы можно загружать из и записывать в ASCII файлы.

Тестовый диалог (Test Dialog) используется для отладки и трассировки SCIL-программ и выражений, а также для просмотра значений переменных различных типов. Настройки тестового диалога сохраняется в профиле пользователя.

3.8.

Периферийное оборудование

Принтеры

Вплоть до 20 принтеров может быть подключено к базовой системе – напрямую или через LAN. Принтеры могут быть различных типов. Кроме того, любой принтер определенный в операционной системе может быть доступен MicroSCADA.

Каждый принтер снабжается уникальным номером, с которым связывается определенная задача – например, печать тревог и событий, печать «твердых» копий экранов, ретроспективных отчетов и т.п. Принтер можно запрограммировать на автоматический «подхват» задач другого принтера.

Печать может выполняться автоматически либо инициироваться вручную. Пользователь может форматировать разметку печати. Основные виды печати – списки событий, отчеты, «твердые» копии экранов и документы. Автоматическая печать списков событий активизируется событиями процесса и может выполняться одним или несколькими принтерами.

Аварийная сигнализация

Система MicroSCADA может оснащаться устройством аварийной сигнализации, которое представляет собой плату, устанавливаемую в главный компьютер системы либо рабочую станцию. Плата оснащена двумя выходами. Один из них подключается к панели аварийной сигнализации (состоящей из ламп и клавиш квитирования). Второй выход может использоваться для соединения с другими устройствами (до 8 штук).

4. Система связи

4.1. Общая информация

Для успешного функционирования системы должна быть организована надежная передача информации между системным сервером SYS 500 и устройствами нижнего уровня. В некоторых случаях, эта информация должна быть также передана на верхний уровень диспетчерского управления, например в ОДУ энергосистемы. Соответственно, команды, получаемые с верхнего уровня, должны быть ретранслированы низовым устройствам. Передача данных - основная задача системы связи. В ходе этого процесса очень часто возникает необходимость преобразования протоколов связи, поскольку протоколы для связи с устройствами нижнего уровня, как правило, отличны от протоколов для связи с верхним уровнем.

4

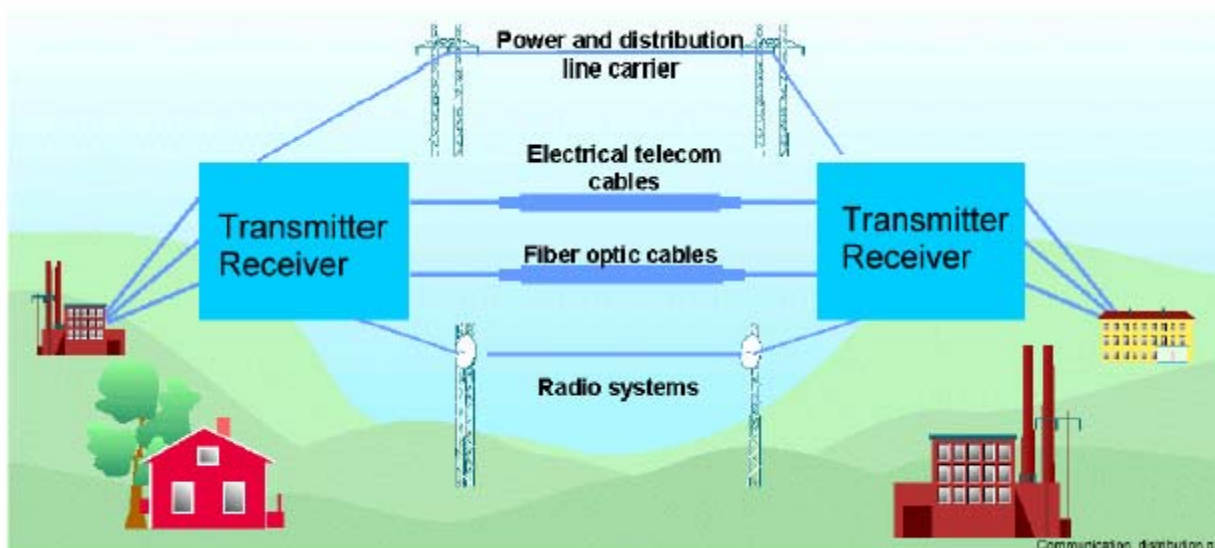


Рис. 4.1.-1 Связь в энергетике

Система связи реализует взаимодействие между различными устройствами в системе MicroSCADA, например, между двумя системными серверами или между двумя серверами связи.

Связь может осуществляться в двух направлениях – с нижним и верхним уровнями (Рисунок 4.1.-2.).

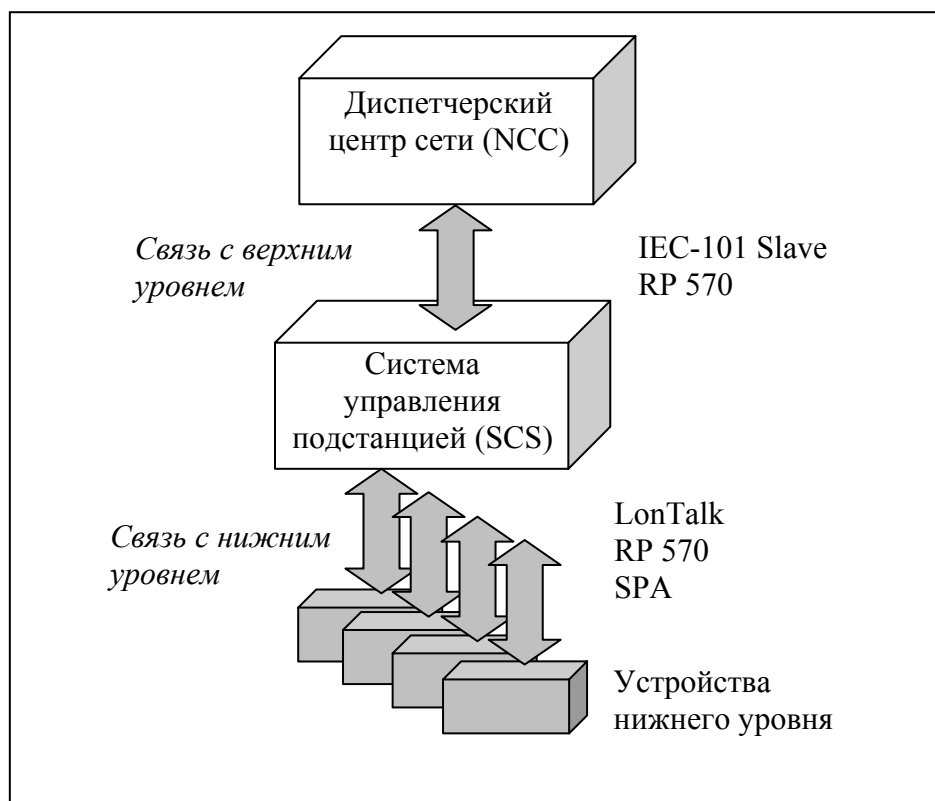


Рис. 4.1.-2 Связь между уровнями

4.1.1

Связь с верхним уровнем

Под связью с верхним уровнем понимается вертикальная связь между устройствами нижнего уровня, системой управления подстанцией (SCS) и диспетчерским центром управления сетью. В диспетчерском центре может быть установлен SYS 500 либо система другого производителя.

В качестве физической среды каналы связи с верхним уровнем чаще всего используют телефонные линии, радио- и ВЧ-связь.

4.1.2.

Связь с процессом

Под связью с процессом понимается связь между системой управления подстанцией (SCS) и терминалами защиты и управления, физически подключенными к процессу энергоснабжения. В случае отсутствия на подстанции такой системы, терминалы защиты и управления подключаются к центру управления сетью напрямую, обычно посредством концентраторов или удаленных терминалов (RTU). Сами терминалы защиты и управления нижнего уровня также объединяются в сеть, называемую промышленной/полевой сетью/шиной.

Поскольку промышленная сеть располагается в непосредственной близости от первичного оборудования, для исключения влияния электромагнитных помех, в качестве физической среды передачи

данных наиболее часто используется оптоволокно. Вследствие большого количества передаваемых данных, для линий связи нижнего уровня, как правило, применяются более быстрые каналы, чем для линий связи с верхним уровнем.

4.1.3.

Сетевая топология

Большинство реализованных в системе MicroSCADA протоколов функционируют в режиме ведущий-ведомый (master-slave). Ведущий (master) – это первичная станция, запрашивающая данные и посылающая команды. Ведомый (slave) – это вторичная станция, передающая данные (по запросу или спонтанно) и исполняющая полученные от ведущего команды.

Некоторые протоколы, например DNP v3.00, поддерживают также режим связи, называемый равноправным (peer-to-peer), когда ведомые напрямую могут взаимодействовать между собой. Количество ведущих и ведомых, подключенных к одной линии связи, ограничивается используемым протоколом. На практике, ограничения диктуются требованиями к быстродействию и времени отклика. Чем больше устройств подключено к линии, тем шире должна быть полоса пропускания используемого канала связи.

Существует несколько возможностей соединения устройств между собой. Соединение точка-точка (point-to-point) может быть реализовано при помощи прямого кабельного соединения, двух узловой радиосвязи либо посредством модемной сети. Топология последовательной шины (multi-drop) может строиться посредством последовательного соединения модемов (вход-выход) или при помощи концентратора. Возможные топологии показаны на Рисунке 4.1.3.-1.

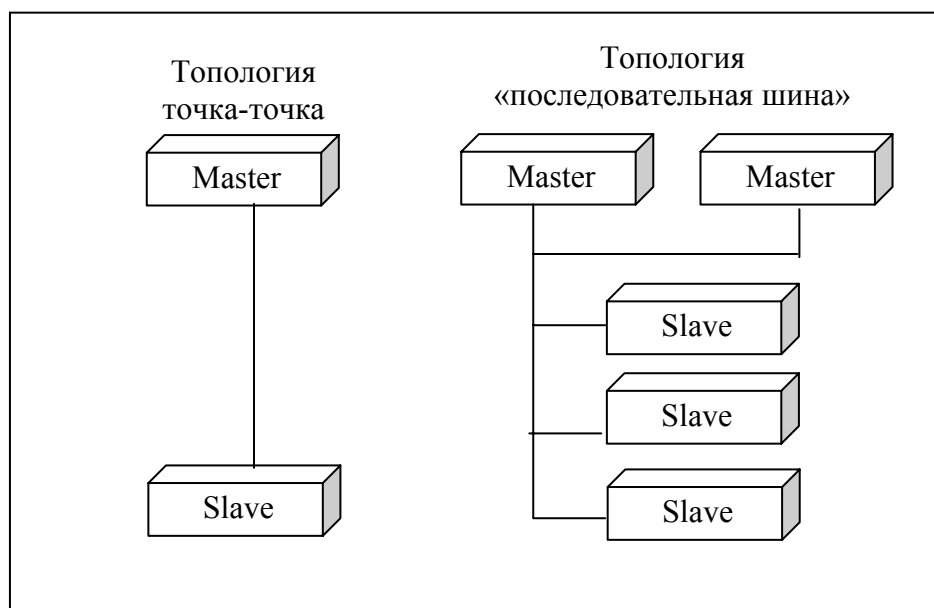
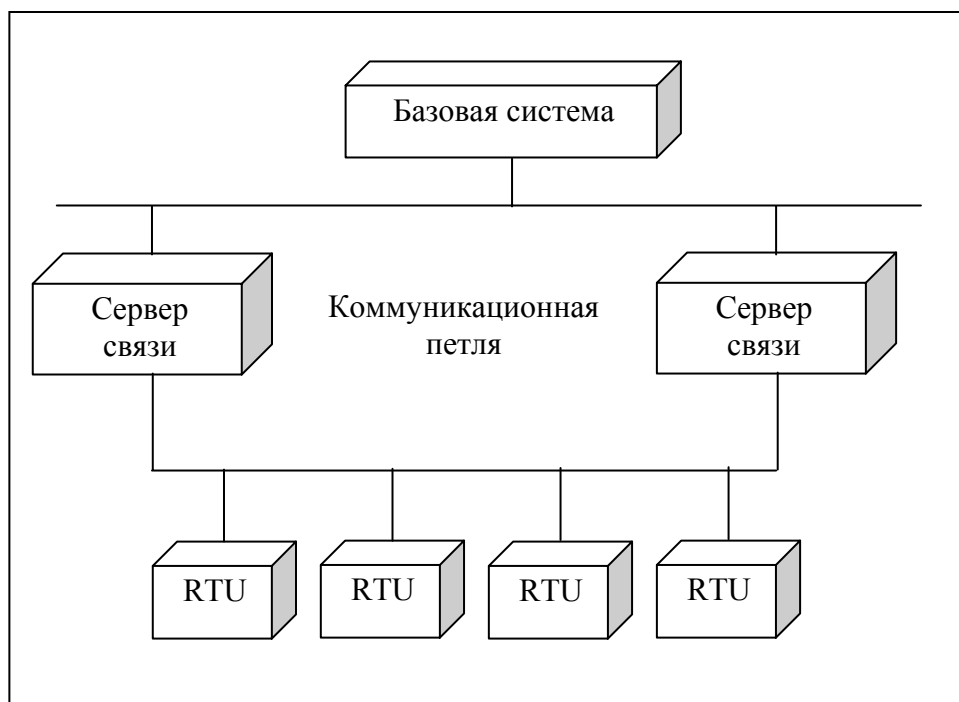


Рис. 4.1.3.-1 Топология точка-точка и последовательная шина

Необходимо выделить особый режим работы последовательной шины, именуемый коммуникационной петлей (Рисунок 4.1.3.-2).

Коммуникационная петля образуется посредством линий связи. Это означает, что в случае повреждения одной из линий, всегда существует другая, ведущая к этому же узлу. Резервирование достигается без необходимости дублирования каждого канала связи.



4

Рис. 4.1.3.-2 Коммуникационная петля – вид многоабонентской линии, при котором каждый узел имеет альтернативное подключение

4.2.

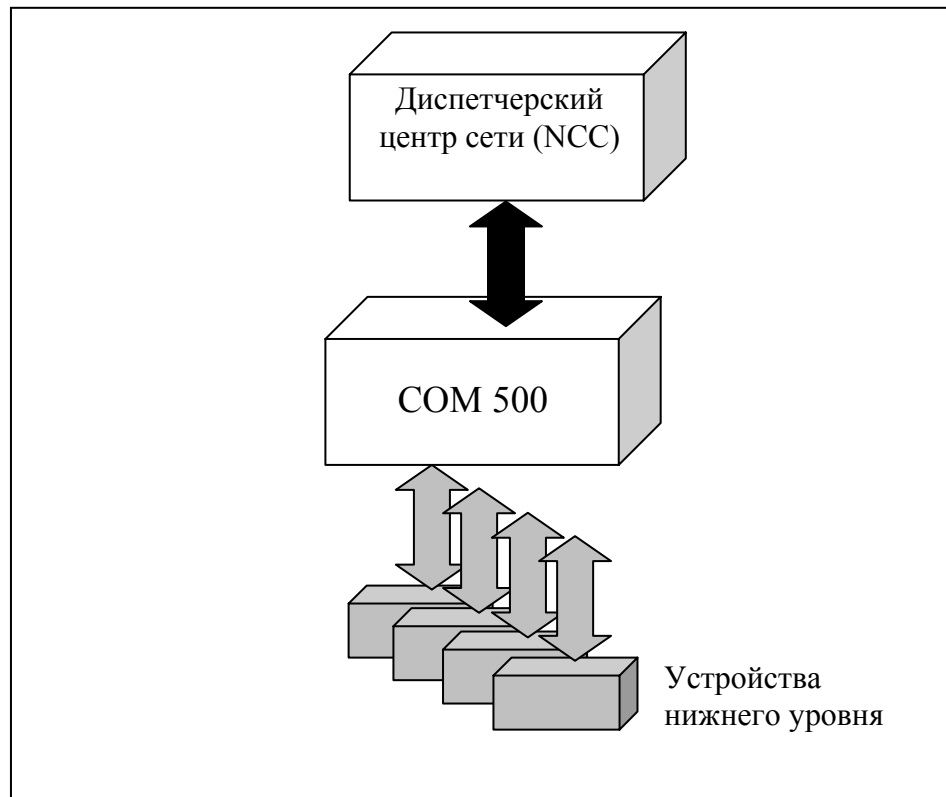
Сервер связи COM 500

COM 500 – это коммуникационный сервер, предоставляющий услуги межсетевое интерфейса для маршрутизации потоков данных между процессом и верхними уровнями системы. Передача данных, как правило, требует преобразования протоколов. Кроме того, COM 500 занимается координацией общесистемных задач, таких как оперативная авторизация управляющих команд (Рисунок 4.2.-1).

COM 500 поддерживает большое количество протоколов для соединения с системами верхнего уровня. Диспетчерские центры обычно подключаются посредством телемеханических протоколов, например RP 570. К COM 500 могут подключаться различные устройства нижнего уровня, такие как LONWORKS -устройства, SPA-устройства и RTU. COM 500 посылает им информацию для контроля и управления первичным процессом.

Также как SYS 500, COM 500 базируется на технологии MicroSCADA. COM 500 может быть интегрирован в системный сервер для получения

компактных и дешевых решений. COM 500 оснащен диагностическим интерфейсом для контроля состояния системы связи.



4

Рис. 4.2.-1 COM 500 – межсетевой интерфейс для передачи данных между устройствами процесса и верхним уровнем

Функциональное окружение

Рисунок 4.2.-2 демонстрирует функциональное окружение сервера связи COM 500.

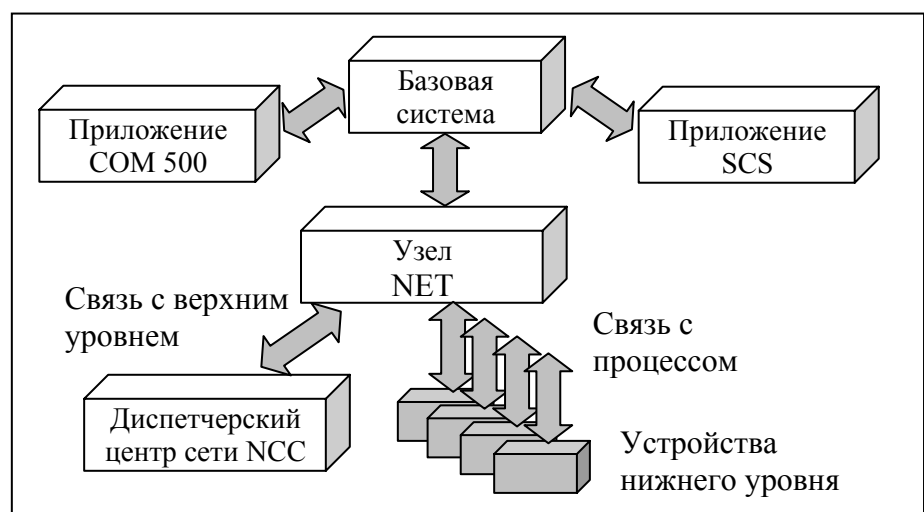


Рис. 4.2.-2 Модель окружения COM 500

Пример системы

COM 500 – это сервер связи, исполняемый в базовой системе MicroSCADA. Он может использовать узлы связи DCP-NET и PC-NET. Несколько COM 510 и COM530 могут быть подключены к COM 500 для увеличения производительности системы. Пример системы на базе COM 500 приведен на Рисунке 4.2.-3.

4

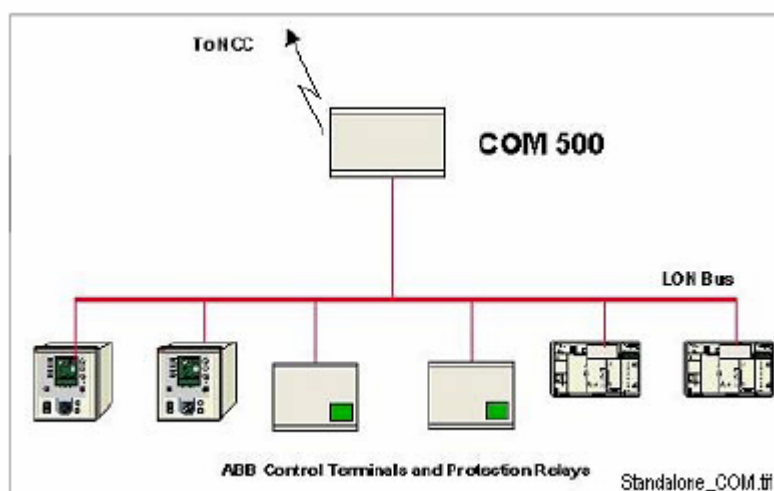


Рис. 4.2.-3 Пример системы COM 500

COM 500 «видит» каждый диспетчерский центр NCC и локальный модуль процесса как станцию (локальные станции представляются объектами системного типа). Связь осуществляется через базу данных процесса.

Связь между COM 500 и NCC реализуется командными процедурами. Командные процедуры посылают информацию в узлы NET для преобразования и передачи. Обычно, для обработки одного типа данных предназначается отдельная командная процедура. Передача данных всегда активизируется после старта системы и восстановления работоспособности каналов связи. В процессе работы системы события процесса запускают командные процедуры через каналы событий.

Команды и уставки, посылаемые из NCC на подстанцию, доставляются в приложение COM 500 через объекты процесса. Объекты процесса активизируют каналы событий для запуска процедур обработки полученных команд. Для обеспечения безопасности используются команды с подтверждением выбора и двухшаговый механизм авторизации.

4.3.

Выделенные серверы связи

Выделенный сервер связи – это компьютер, специально зарезервированный для функции связи с процессом. Он подключается к системному серверу через локальную сеть (LAN). Выделенный сервер применяется для увеличения числа линий связи, для доступа к процессу через локальную сеть и снижения вычислительной нагрузки на системный сервер SYS 500.

4

Сервер связи может содержать узлы PC-NET и DCP-NET. Обычно сервер связи оснащается интерфейсом оператора. Примеры системных компонентов сервера связи показаны на Рисунке 4.3.-2. К ним относятся программное обеспечение (ПО) DCP-NET, DCP-плата, COM-порты и MFL.

MFL – базовое ПО сервера связи типа COM 510 и COM 530. Оно используется для загрузки программ DCP-NET в платы DCP и поддержки связи между базовой системой и узлами NET.

Выделенный сервер COM 510

COM 510 – это компьютер, работающий под управлением DOS и содержащий от 1 до 4 DCP-NET и MFL. Необходимое аппаратное обеспечение состоит из платы DCP (Emulex). COM 510 подключается к базовой системе посредством локальной сети. Базовая система может размещаться в SYS 500 либо COM 500. См. Рисунок 4.3.-1.

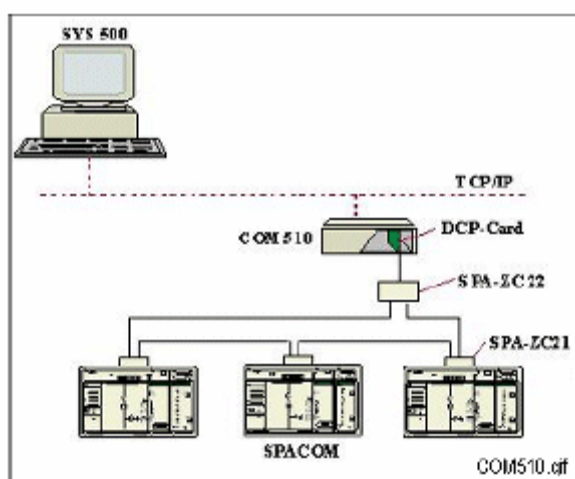


Рис. 4.3.-1 Пример конфигурации COM 510

Выделенный сервер COM 530

COM 510 – это компьютер, работающий под управлением Windows NT/2000, содержащий от 1 до 2 DCP-NET и/или PC-NET, базовую систему и выделенный для маршрутизации сообщений. См. Рисунок 4.3.-2.

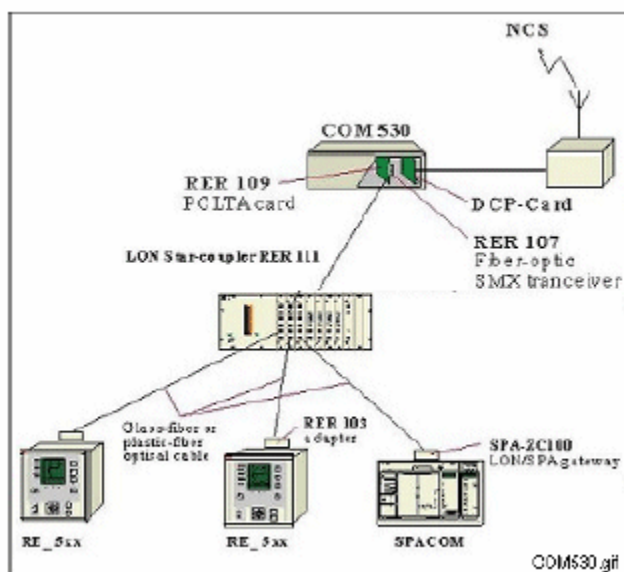


Рис. 4.3.-2 Пример конфигурации COM 530

4.4.

Узел NET

Узел NET состоит из программного обеспечения NET и аппаратного обеспечения для подключения к каналам связи. Узлы NET могут размещаться в базовом компьютере либо в сервере связи. Например, ПО PC-NET и плата PCLTA рассматриваются как один узел NET. Кроме того, узел NET включает в себя файлы конфигурации самого NET, линий связи и подключенных к ним устройств. К узлам NET могут подключаться принтеры тревог и событий.

Узлы NET применяются для увеличения числа доступных протоколов, COM-портов, а также для снижения загрузки процессора и потребления памяти основного компьютера системы.

Задачи

Задача узла NET – преобразование внешних протоколов в формат внутреннего протокола MicroSCADA, который используется для доставки сообщений между компонентами системы. Внешние протоколы применяются для связи с устройствами процесса. Таким образом, к MicroSCADA могут подключаться различные устройства нижнего уровня.

В дополнение к этому, узлы NET обрабатывают следующие задачи:

- Маршрутизацию сообщений между физическими устройствами нижнего уровня и приложениями в базовой системе
- Диагностический контроль подключенных устройств и линий связи

Типы узлов NET

В MicroSCADA применяются узлы NET двух типов – DCP-NET и PC-NET. DCP-NET состоит из программного обеспечения DCP-NET и специфической платы Emulex, тогда как PC-NET – это программа, исполняемая процессором базового компьютера.

Узел DCP-NET

DCP-NET – это узел NET, состоящий из ПО связи DCP-NET и платы DCP. Узлы DCP-NET могут быть внутренние и внешние. Внутренний узел находится в базовом компьютере, это может быть COM 500 или SYS 500. В один компьютер можно установить до двух DCP-плат.

Внешний узел NET – это выделенный сервер связи. В нем может быть установлено до 4 DCP-плат. Каждая плата содержит 8 линий связи.

Узел PC-NET

Программное обеспечение PC-NET похоже на DCP-NET. PC-NET выполняется процессором главного компьютера параллельно с ядром MicroSCADA. PC-NET может находиться в базовом компьютере либо в сервере связи.

Для связи по протоколу LonTalk® PC-NET использует платы PCLTA (PC LonTalk адаптеры). Кроме того, для PC-NET используются COM-порты. В случае, если компьютер не оснащен достаточным количеством COM-портов, применяются платы-расширители (RocketPorts).

4.5. Компоненты системы связи

Система связи включает программное обеспечение, аппаратное обеспечение и сети.

4.5.1. Программное обеспечение связи

Программное обеспечение системы связи состоит из протоколов, их программного окружения, ПО поддержки и операционной системы.

Протоколы

Передача данных, кроме физических каналов связи, требует также наличия логических соглашений, называемых протоколами. Протокол – это своеобразный набор «правил дорожного движения» и договоренностей, необходимых для передвижения информации по сети. Они определяют содержание, формат, синхронизацию, разбиение и обработку ошибок при передаче сообщений.

Передаваемые данные кодируются в соответствии с правилами используемого протокола. Протоколы являются одной из важнейших частей системы.

SPA, RP 570 и LonTalk – примеры «мастер»-протоколов для связи с процессом в системе MicroSCADA. Связь между базовой системой и серверами связи базируется на внутреннем протоколе MicroSCADA. Протоколы в режиме «ведомого» используются для связи между узлами NET и диспетчерскими центрами.

4

ПО разработки протоколов

Программный интерфейс связи (CPI) – прикладное программное обеспечение для реализации протоколов в среде MicroSCADA. CPI представляет собой набор функций, при помощи которых осуществляется преобразование сообщений внутреннего протокола MicroSCADA в другие форматы. Прикладные программы, обменивающиеся данными с MicroSCADA из других систем, могут быть созданы посредством CPI. Связь между такими программами и MicroSCADA происходит по стеку TCP/IP.

ПО поддержки

ПО поддержки необходимо для загрузки и выполнения ПО связи. Это может быть базовая система либо MFL. Базовая система используется для поддержки PC-NET.

Операционная система и ПО устройств

Используемые операционные системы представлены DOS и Windows NT/2000. В SYS 500 и COM 500 используются только Windows NT/2000. ПО устройств нижнего уровня состоит из различных программ для обработки связи с устройствами LMK, SPA или принтерами. Для передачи сообщений это ПО использует описание точек ввода/вывода узлов NET, например LONWORKS или SPA.

4.5.2.

Сети

Для организации связи в MicroSCADA используются локальные сети LAN, сети LONWORKS и последовательные линии.

Локальные сети LAN

Локальные вычислительные сети LAN – это сети, охватывающие географически небольшие районы, например офисная сеть компании. В системе MicroSCADA LAN обычно используются для связи системных серверов, серверов связи, рабочих станций операторов и систем верхнего уровня. Локальные сети обладают большой емкостью каналов связи. Для передачи данных применяется стек протоколов TCP/IP.

Сети LONWORKS

Сеть LONWORKS – это сеть, где для передачи данных используется протокол LonTalk. При помощи технологии LONWORKS возможно построение распределенных систем защиты и контроля с более чем одним центром диспетчерским управлением NCC. NCC может получать измерения, маркированные временем события и статусные коды от устройств процесса. Сеть LONWORKS строится при помощи концентраторов и имеет топологию «звезда».

4

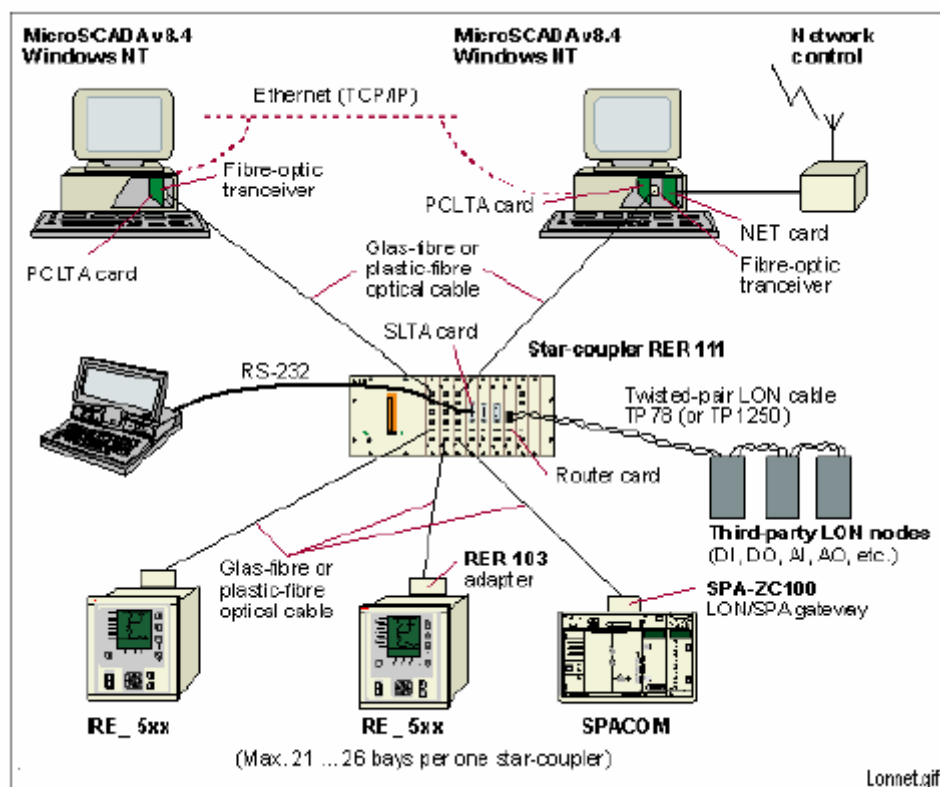


Рис. 4.5.2.-1 Пример сети LONWORKS

Последовательные линии

Последовательная передача данных может использоваться как для связи с верхним уровнем, так и для связи с устройствами процесса. Обычно это происходит в случае невозможности использования локальной сети. Последовательные линии связи подключаются к COM-портам компьютера.

4.5.3.**Аппаратное обеспечение системы связи**

Аппаратное обеспечение системы связи включает аппаратное обеспечение протоколов, коммуникационные платы и внешнее аппаратное обеспечение.

Аппаратное обеспечение протоколов

Аппаратное обеспечение протоколов необходимо для загрузки и выполнения ПО протоколов. К этой категории относятся PC и DCP-платы.

Коммуникационные платы

Коммуникационные платы используются для связи между базовой системой и внешними узлами. Коммуникационные платы размещаются в базовом компьютере – например, PCLTA-плата или плата-расширитель портов RocketPorts. Эти устройства используются программным обеспечением PC-NET.

4

PCLTA-плата – это PCI или ISA плата для подключения базового компьютера к сети LONWORKS. Плата RocketPorts – это PCI-плата для увеличения количества доступных системе последовательных портов.

Внешнее аппаратное обеспечение

Внешнее аппаратное обеспечение – это аппаратное обеспечение связи, не входящее в состав узлов NET. Это могут быть модемы, сетевые интерфейсные карты, переключатели линий или устройства синхронизации времени.

Модем – это устройство для преобразования цифровых сигналов в аналоговые. Одна модемная линия связи требует наличия двух модемов по обоим концам.

Сетевые интерфейсные карты используются для подключения к сетям, например к локальной вычислительной сети предприятия (LAN). Для резервированного подключения требуются карты специального типа.

Переключатель линии – это устройство для физического переключения между последовательными линиями связи, логически подключенными к одному COM-порту. Может использоваться в системах с горячим резервированием.

Под синхронизацией времени понимается синхронизация внутренних часов MicroSCADA от внешнего источника. Таковым может служить:

- GPS (Global Positioning System) – система спутниковой навигации

- Радиочасы
- Устройство на верхнем уровне системы, например сервер NCC

Часы в базовом компьютере и серверах связи всегда синхронизируются между собой

4.6.

Устройства процесса

Устройства процесса – это устройства нижнего уровня системы, физически подключенные к процессу электроснабжения. Они собирают данные о протекании процесса и исполняют команды, поступающие от базовой системы. Примерами устройств процесса являются REx-, SPA-, LMK-устройства и RTU. В текущей версии системы MicroSCADA все эти устройства представляются станциями, предыдущие версии базировались на концепции RTU.

Устройства процесса разделяются на первичные и вторичные. К первичным устройствам относится силовое оборудование – выключатели, разъединители и т.д. Вторичные устройства применяются для автоматизации, защиты и управления силовым оборудованием (Рисунок 4.6.-1).

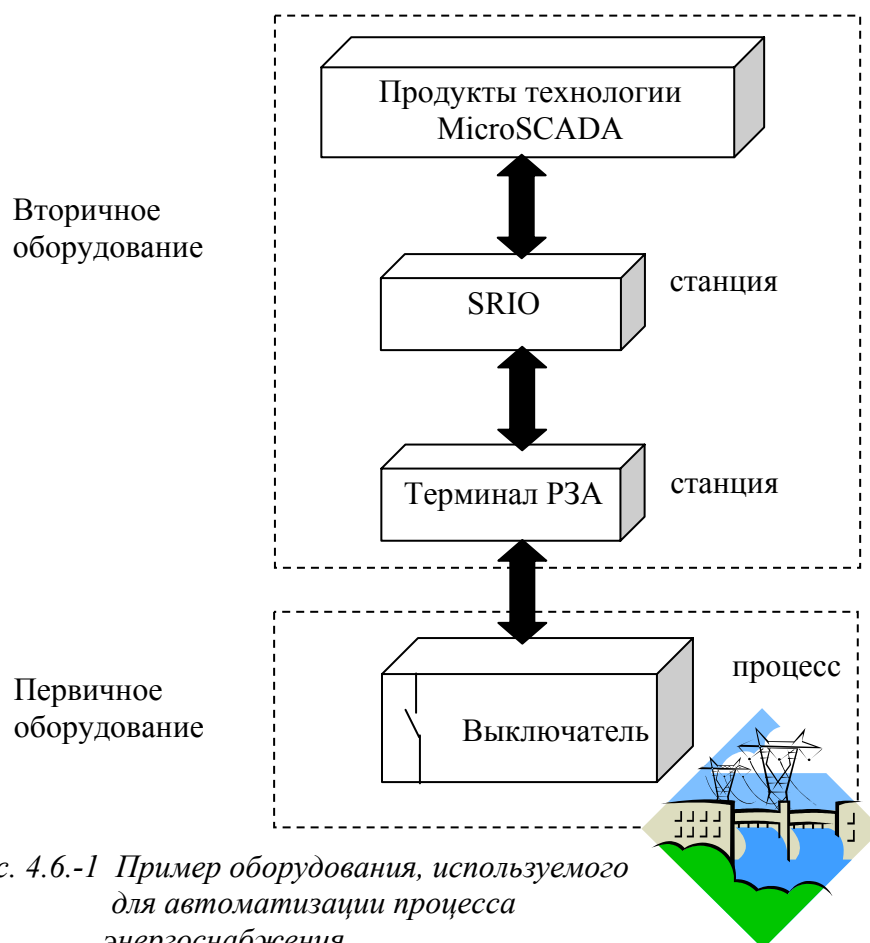


Рис. 4.6.-1 Пример оборудования, используемого для автоматизации процесса энергоснабжения.

5. Прикладные библиотеки LIB 5xx

5.1. Разработка приложения

Процесс разработки приложения заключается в реализации специфических функций и алгоритмов управления. Приложение содержит изображения, диалоги, SCIL-программы и необходимые описания объектов (Рисунок 5.1.-1).

5

Для создания приложения используются редактор изображений, инструмент определения объектов (Object Definition Tool), прикладные библиотеки и программирование на SCIL. Прикладные библиотеки ускоряют процесс инжиниринга за счет использования стандартных решений, тогда как программирование позволяет реализовывать нестандартные функции, но требует больше времени и высокой квалификации.

Процесс разработки приложения включает в себя следующие этапы:

- Создание интерфейса пользователя (редактор изображений, редактор диалогов, прикладные библиотеки). Инсталляция и конфигурирование стандартных функций выполняется в редакторе изображений.
- Создание и определение объектов приложения. Объекты приложения определяются при помощи редактора изображений либо инструмента определения объектов (Object Definition Tool).
- Программирование расширенной функциональности изображений. Выполняется посредством редактора SCIL.
- Программирование расширенной функциональности приложения, например самодиагностики или расчетов по измерениям.

Эти задачи обычно выполняются параллельно. Поскольку на одном экране можно одновременно отобразить несколько мониторов MicroSCADA с различными изображениями, существует возможность осуществлять управление процессом, инжиниринг приложения и контроль над системой в одно и то же время.

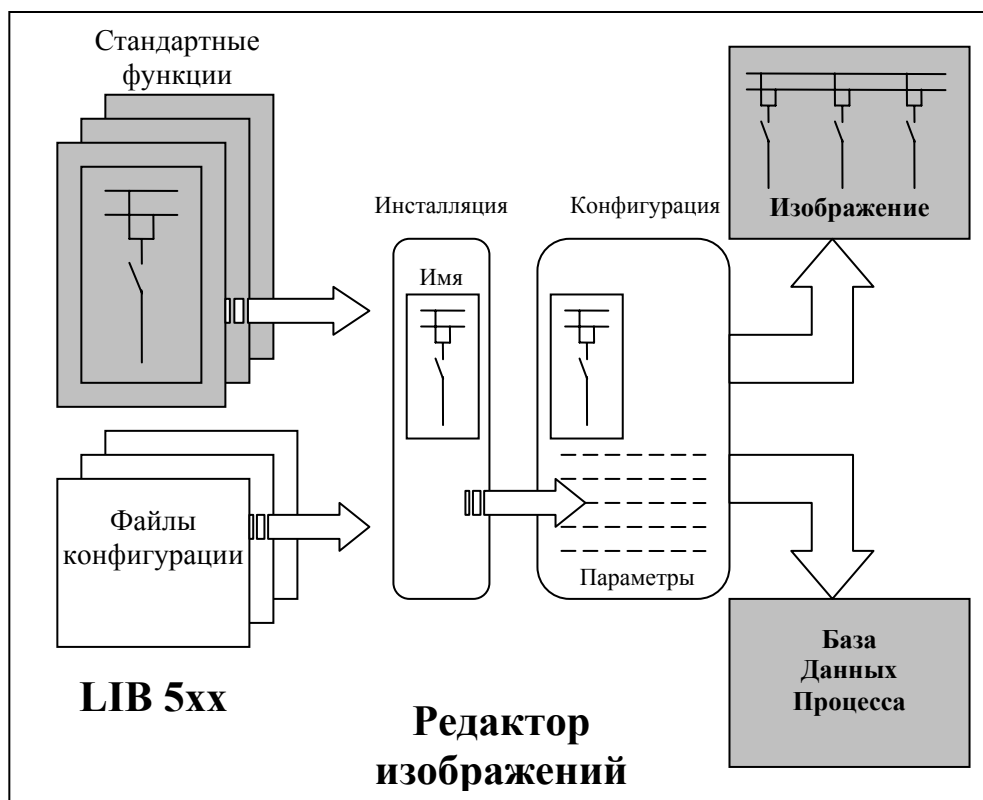


Рис. 5.1.-1 Создание приложения при помощи стандартных функций

5.2.

Прикладные библиотеки

Прикладные библиотеки – это продукты технологии MicroSCADA, представляющие собой наборы стандартных функций приложения. Прикладные библиотеки существенно ускоряют процесс инжиниринга приложения. Они содержат готовые функции, устанавливаемые в приложение. Связанные с функциями объекты создаются автоматически во время инсталляции. Атрибутам созданных объектов присваиваются значения «по умолчанию».

Приложение может быть создано путем инсталляции и конфигурирования функций из прикладных библиотек, например LIB 510.

Прикладные библиотеки LIB 5xx

LIB 5xx – это набор прикладных библиотек, включающий в себя следующие продукты:

- LIB 500 Base (базовая) – платформа для поддержки других пакетов (LIB 5xx), содержит некоторые базовые функции: список тревог, список событий, динамическую окраску шин и систему самодиагностики.
- LIB 510 – пакет функций для поддержки процесса среднего напряжения

- LIB 520, LIB 530, LIB 580 – пакеты для поддержки процесса высокого напряжения

Выбор используемой библиотеки определяется установленными в системе устройствами процесса. Каждый производитель оборудования АBB предоставляет прикладную библиотеку для поддержки своих устройств автоматизации.

В состав стандартных функций включены программные инструменты для задания уставок и выгрузки осциллограмм из терминалов защиты и управления. Они могут запускаться из динамической мнемосхемы подстанции. Например, в состав LIB 510 входят функции поддержки устройств SPACOM, RED 5xx и RTU.

5

Стандартные функции копируются из прикладных библиотек и конфигурируются в соответствии с условиями конкретного проекта. Таким образом, удается сэкономить большое количество времени, поскольку все эти функции прошли заводское тестирование и обладают высокой надежностью. Другая немаловажная особенность – постоянная поддержка и обновление программного обеспечения производителем, что гарантирует его соответствие аппаратным версиям оборудования.