

Perfection in Automation
www.br-automation.com



Разработка проектов TM830

Версия: V1.10 / версия: 1 января 2008
Номер модели: TM830TRE.30-ENG

Требования:	
Учебные программы	TM800 – <i>APROL</i> Концепция системы
Программное обеспечение	SuSE LINUX
Аппаратное обеспечение	1 управляющий компьютер, 1 контроллер

Мы имеем право вносить изменения в содержание настоящего документа без предупреждения. Информация, содержащаяся в настоящем документе, считается достоверной на дату публикации, однако компания Bernecker + Rainer Industrie-Elektronik Ges.m.b.H. не дает никаких гарантий, явно выраженных или подразумеваемых, по отношению к продуктам или документации, содержащейся в настоящей книге. Помимо этого, компания Bernecker + Rainer Industrie-Elektronik Ges.m.b.H. не несет ответственности в случае случайного или косвенного ущерба, причиненного в связи с или в результате поставки, работы или использования указанных продуктов. Названия программных и аппаратных продуктов, товарных знаков, используемых в настоящем документе, зарегистрированы соответствующими компаниями.

Оглавление

1	ВВЕДЕНИЕ	5
1.1.	Цели	6
2.	РАЗРАБОТКА ПРОЕКТОВ	7
3.	УПРАВЛЕНИЕ ОСНОВНЫМИ ДАННЫМИ	8
3.1.	Основные параметры проекта для разработчиков	15
3.2.	Профили проектов	17
4.	ФИЗИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ	
4.1.	Создание дерева физического представления	22
5.	СТРУКТУРИЗАЦИЯ ПРОЕКТА	
5.1.	Вставка каталогов	25
6.	КОНФИГУРАЦИЯ ОБОРУДОВАНИЯ	27
6.1.	Вставка контроллера	27
6.2.	Настройка контроллера	29
6.3.	Настройка стойки Ethernet POWERLINK	47
7.	КОНФИГУРАЦИЯ УПРАВЛЯЮЩЕГО КОМПЬЮТЕРА	51
7.1.	Добавление управляющего компьютера	51

7.2. Конфигурация управляющего компьютера	52
8. КОНФИГУРИРОВАНИЕ КОНТРОЛЛЕРА С ПРОСТОЙ ЛОГИКОЙ	63
8.1. Общая информация о структурных схемах	63
8.2. Создание структурной схемы	64
8.3. Свойства структурной схемы	66
8.4. Навигация в редакторе структурных схем	68
8.5. Функциональные блоки	70
8.6. Подключения	77
8.7. Схемы I/O	78
8.8. Активирование и компиляция структурной схемы	83
9. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В APROL	86
10. СОЗДАНИЕ И РАСПРОСТРАНЕНИЕ	88
10.1. Версии	88
10.2. Активация	88
10.3. Компиляция	90
10.4. Утверждение	92
10.5. Создание	92
10.6. Пункт меню – повторная компиляция	95
10.7. Загрузка	97

11. ДИСПЕТЧЕР ЗАГРУЗКИ	102
12. ОТЛАДКА СХЕМЫ	
12.1. Заданные значения	110
13. ПЕРЕКРЕСТНЫЕ ССЫЛКИ	111
14. СОЗДАНИЕ ПРИЛОЖЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЦЕССОМ	
14.1 Создание диаграмм процессов	113
15. ГИПЕР-МАКРОСЫ / СТАНДАРТНЫЕ ЭЛЕМЕНТЫ	
15.1. Что такое гипер-макрос?	120
16. КРАТКИЕ ВЫВОДЫ	125
Список контрольных вопросов (короткие вопросы по разделам)	126
16.2. Ответы на контрольные вопросы	126
17. ПРИЛОЖЕНИЕ	128
17.1. Журнал версий	128

1. ВВЕДЕНИЕ

Проектирование составляет большую часть работ, необходимых для создания работоспособных систем. При проектировании важно с самого начала работ определить объем данных по проекту и логику, которую необходимо реализовать.

Подобные работы обычно выполняют в соответствии с функциональными требованиями, которые после уточнения всех параметров в *APROL* ложатся в основу спецификации на проектирование программного обеспечения. Указанные задачи возлагаются на заказчика и руководителя проекта.

Сразу же после выполнения перечисленных выше задач следует этап успешного проектирования.

Рис. 1.
Планирование проекта

CaeManager

Perfection in Automation
www.br-automation.com



ОТЛАДКА СХЕМЫ
ПЕРЕКРЕСТНЫЕ ССЫЛКИ

СОЗДАНИЕ ПРИЛОЖЕНИЯ СИСТЕМЫ
УПРАВЛЕНИЯ ПРОЦЕССОМ

ГИПЕР-МАКРОСЫ / СТАНДАРТНЫЕ ЛИ

Рис. 2. Общее представление

Система
проектирования
APROL

2. ПРОЕКТИРОВАНИЕ

Перед началом создания проекта необходимо определить требования к нему. Определяются размер проекта в зависимости от количества станций оператора, точек ввода / вывода и способа их структурирования, а также основные принципы функционирования, включающие количество прав доступа, цвета и шаблоны.

После принятия решения о структуре выбираются наиболее подходящие компоненты системы управления процессом, такие как серверы, станции операторов и контроллеры. Определяются шина обработки данных, ее топология, количество контроллеров и структура уровня ввода / вывода.

Однако отсутствие заранее известной конфигурации системы не является существенным препятствием, т.к. всегда присутствует возможность расширения системы.

3. УПРАВЛЕНИЕ ОСНОВНЫМИ ДАННЫМИ

При создании нового проекта из пункта меню «**File / New**» (файл/ новый) открывается диалоговое окно «**Project properties**» (свойства проекта). В данном окне содержатся дополнительные подгруппы.

Свойства существующего проекта можно открыть щелчком правой кнопки мыши на папке проекта и выбором пункта меню «**Properties**» (свойства).

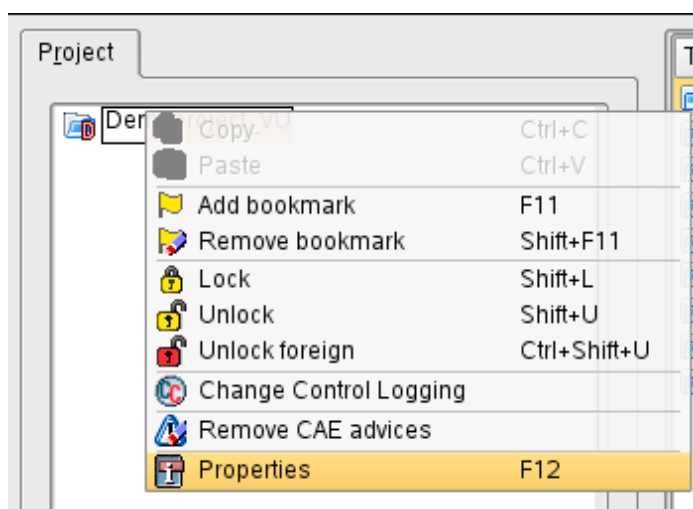


Рис. 3. Свойства проекта

Основные данные проекта

Первая вкладка обычно используется для ввода и управления основными данными проекта. Большинство записей используется только в целях документирования. Допускается хранение таких данных как сведения о компании и логотип. Введенная в данном окне информация позднее появляется при выводе на печать структурных схем.

При помощи указанных данных предоставляется информация о создателе проекта и его заказчике. При этом разработчик снабжается документацией, внешний вид которой соответствует политике компании.

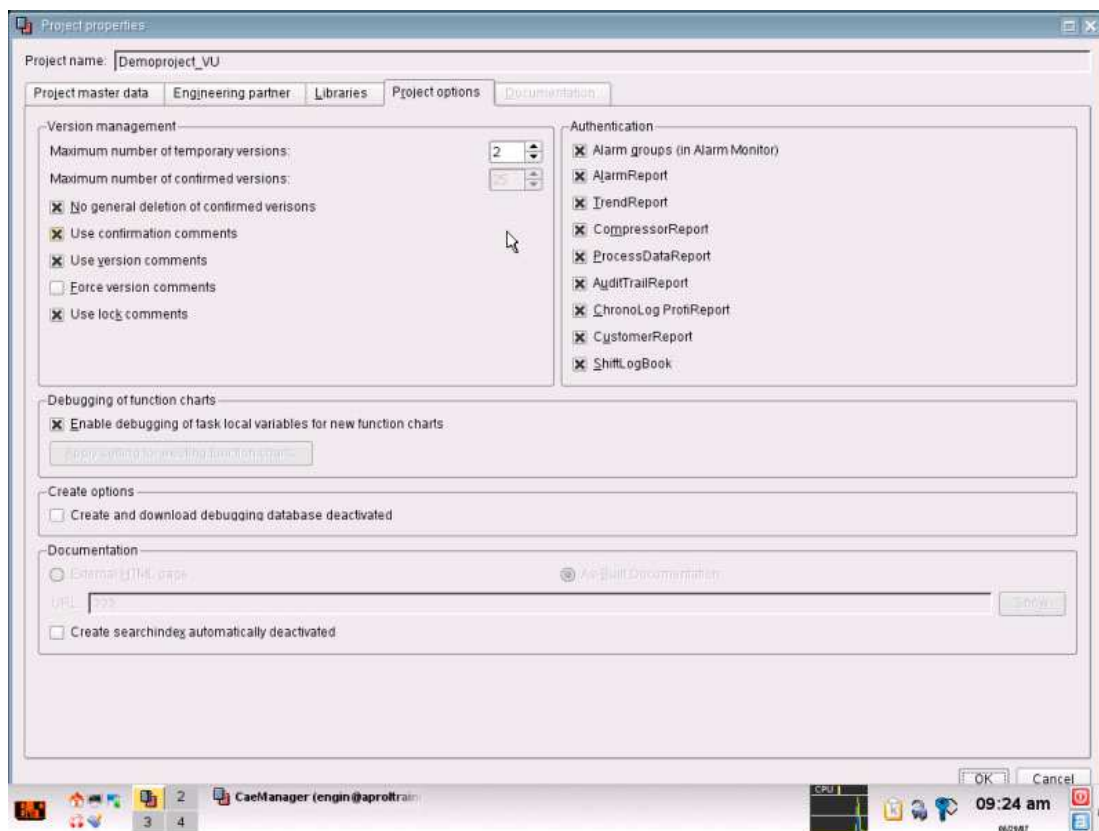


Рис. 4. Основные данные компании

Помимо этого, может быть установлен флаг **«cooperation with engineering partner»** (сотрудничество с партнером по разработке). Установка флага активирует новое окно (Engineering Partner – партнер по разработке), в котором при необходимости могут вводиться данные.

Библиотеки:

С системой APROL поставляются следующие библиотеки:

- **APROL** Расширяя библиотеки IEC, содержит дополнительные блоки, включая функциональные блоки системы управления для настройки **систем сигнализации, трендов** и т.д.
- **IEC611313** Содержит все **функциональные блоки IEC 1131-3**
- **DCS** Содержит **гипер-макросы и типовые элементы** для **приводов, фурнитуры, регуляторов и измеренных значений**. Помимо этого, в библиотеке находится большое количество рабочих

шаблонов и лицевых панелей.

Примечание. Работа

Шаблон оператора или лицевая панель – это окно, связанное с диаграммой процесса, используемое для запуска отдельных устройств или оборудования. Термин возник на начальном этапе развития технологии управления процессами при попытке имитировать лицевые панели, например, аппаратных УЧПУ. Шаг вперед от дискретно выстроенных уровней операторов (диаграмм мозаичных схем) к системам управления означал упрощение работы операторов системы за счет воспроизведения максимально возможного объема предшествующих наработок.

• **SYSTEM** Содержит функциональные блоки и гипер-макросы, позволяющие настраивать **самодиагностику системы**.

• **SYSTEM_RIO** Системный мониторинг удаленных устройств ввода / вывода

- **SysMon2005** Системный мониторинг контроллеров, начиная с 2005 серии (SG4 с OS V285 и выше).

- **IS010628** Содержит технологические карты для систем управления производственными процессами.

Примечание. Библиотеки

В *APROL* допускается создание ряда дополнительных библиотек по проектам или по отраслям. Руководитель проекта может рекомендовать всем остальным пользователям использовать только функциональные блоки и макросы картинок из библиотек текущего проекта или пользовательских библиотек. Обеспечивается высокий уровень безопасности, т.к. доступ и внесение изменений в выбранные библиотеки регулируются правами доступа пользователей и паролями. Дополнительно разработчикам предоставляются возможности для проверки системы.

Примечание. Требуемые библиотеки

С проектом необходимо связывать только требуемые библиотеки.

Ключевая фраза – «столько, сколько необходимо», а не «как можно больше».

Параметры проекта

На тип проектного решения влияют другие записи в «**Project options**» (параметрах проекта). Такие объекты, как структурные схемы, диаграммы процессов, компоненты системы управления процессом и т.д., настраиваемые в APROL, способствуют реализации **системы управления версиями**. Любое изменение одного из указанных объектов требует создания новой версии объекта.

Для управления версиями выбранного проекта важную роль играют следующие записи:

- Maximum number of temporary versions (максимальное количество временных версий):

данное значение определяет количество версий объекта. Если при сохранении версии заданное количество превышает, у пользователя запрашивается удаление существующей версии.

- Maximum number of confirmed version (максимальное количество подтвержденных версий):

если не выбран параметр «**No general deletion of confirmed versions**» (без общего удаления подтвержденных версий), то может быть указано количество подтвержденных версий объекта.

- No general deletion of confirmed versions (без общего удаления подтвержденных версий):
рекомендуется устанавливать этот флаг. Это гарантирует, что подтвержденные версии объекта, которые часто являются проверенными, не будут удаляться. При этом допускается ручное удаление.

Примечание. Подтверждение версий

«Подтверждение» не обрабатывается автоматически системой.
Пользователь

должен делать это самостоятельно. Это дает пользователю с соответствующими правами авторизации возможность

определить версию как «правильно работающую». Версии,

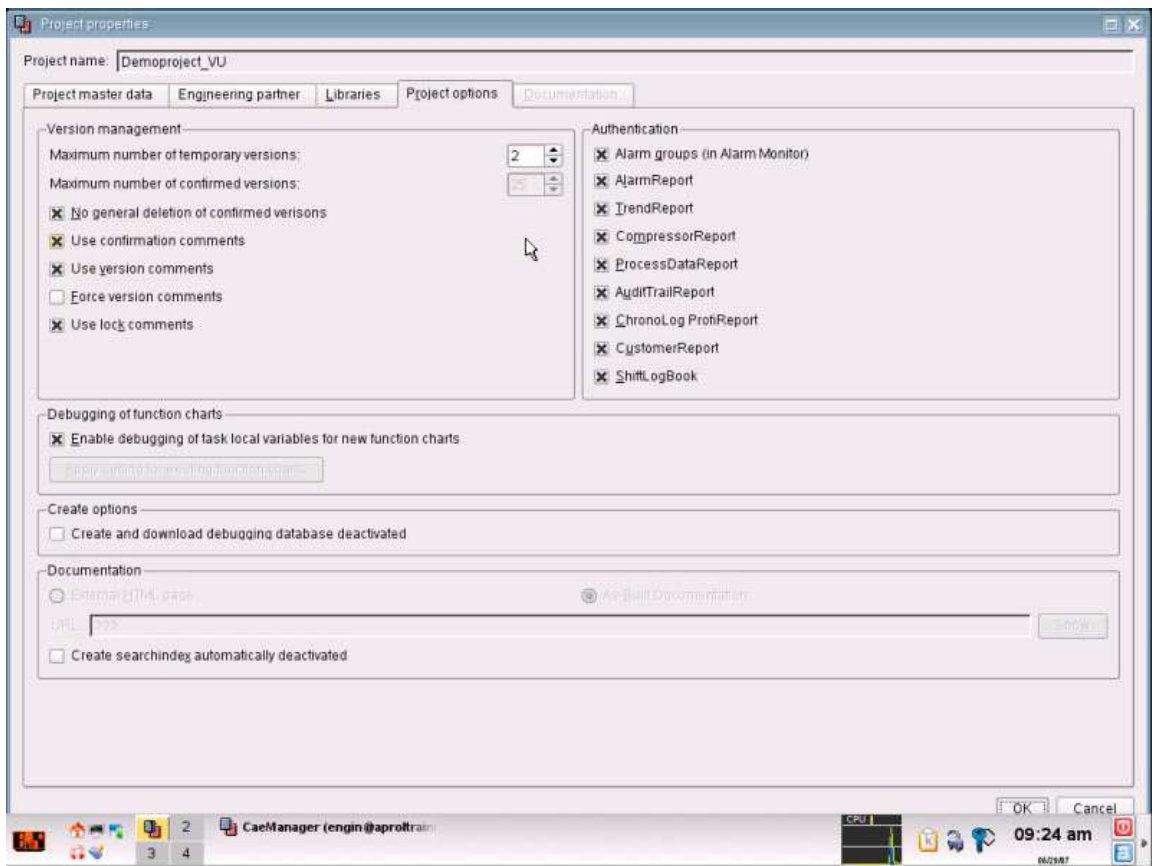
отмеченные подобным образом и (что еще лучше) снабженные в качестве таковых метками,

не должны удаляться.

- Use confirmation comments (использовать комментарии подтверждения):
допускается ввод комментария при подтверждении версии.

- Use version comments (использовать комментарии версий):
аналогично комментариям подтверждения.
- Force version comments (принудительное комментирование версий):
в данном случае должны вводиться комментарии.
- Use lock comments (использовать комментарии блокировки):
добно использовать при работе с заблокированными элементами проекта. Полезно указать пользователя, заблокировавшего элемент проекта, и причину блокировки. Блокировка может быть снята только тем, кто ее установил. Это дает возможность подтвердить заблокированные элементы для автономной разработки.

Рис. 5. Параметры проекта



- Enable debugging of task local variables for new function charts (включение отладки локальных переменных задачи для новой структурной схемы):

если данный флаг установлен, то переменные процесса, которые обычно не передаются по шине обработки данных, отображаются при интерактивной отладке. Хотя данная функция помогает разработчикам обнаруживать ошибки, ее применение несколько увеличивает загрузку шины обработки данных, и функцию следует использовать только в случае крайней необходимости.

Примечание. Отладка

В этом заключается наибольшее различие между программируемыми устройствами, которые обращаются к интерфейсам всех контролеров через интерфейс программируемого устройства, и системой управления процессами, которая может только передавать данные, требуемые другим компонентам системы управления, по шине обработки данных. «Отладка структурной схемы» может использоваться в *APROL* для отображения всех значений структурной схемы.

- Create and download debugging database deactivated (создать и загрузить деактивированную отладочную базу данных):

данный флаг используется для увеличения производительности при генерации проекта. Параметр следует отключить после успешного запуска.

- Create search index automatically deactivated (создать автоматически деактивированный индекс поиска)
- Authentication (проверка подлинности):

управление пользователями может использоваться в следующих областях для указания разрешения или запрета работы с данными приложениями. Требование проверки может быть включено или отключено в рамках всей системы в параметрах проекта.

- Группы аварийных сигналов (в системе контроля аварийных

состояний):

применяется к управлению пользователями в «СРЕДЕ ВЫПОЛНЕНИЯ». За счет этого обеспечивается возможность определения пользователей, которые могут считывать, подтверждать и блокировать группы аварийных сигналов.

- AlarmReport (оповещение об авариях):

применяется при анализе архива аварийных сигналов и уведомлений.

- TrendReport (Отчет по трендам):

применяется при анализе записанных трендов и характеристик кривых.

- CompressorReport (Отчет по сжатым данным):

применяется для анализа сжатых данных (почасовые средние значения, средние значения за день и т.д.).

- ProcessDataReport (Отчет обработки данных):

ко всем переменным процессов, известным в системе APROL.
.

- AuditTrailReport (Отчет по следу контроля):

все действия, выполняемые в СРЕДЕ ВЫПОЛНЕНИЯ или рабочей среде операторов, регистрируются в полном объеме, что позволяет анализировать подобные данные.

- ChronoLogReport (Отчет ChronoLog):

анализируются все данные, записанные через ChronoLog.

- Customer report (Отчет заказчика):

отчет создается заказчиком. Допускается аутентификация внешних отчетов.

- ShiftLogBook (Журнал учета смен):

журнал учета смен используется вместо журнала обслуживания системы.

3.1 Основные параметры проекта для разработчиков

Основные параметры проекта, задаваемые разработчиками, определяются в пункте меню **«Extras / Options ...»** (дополнительные возможности / параметры).

Основные параметры продукта предоставляют разработчикам, вошедшим в систему, возможность настройки. Разработчику предоставляются параметры настройки для работы с CaeManager (запросы, меню и т.д.). Эти параметры загружаются при каждом входе разработчика в систему, что позволяет определенным образом настраивать сообщения подтверждений, появляющиеся в системе.

^

Предоставляется несколько возможностей настройки. Для получения дополнительной информации см. интерактивные справочные файлы APROL.

Разработчик может определить, будут ли отображаться или запрещаться системой сообщения подтверждений, перечисленные на вкладке **«confirmations»** (подтверждения).

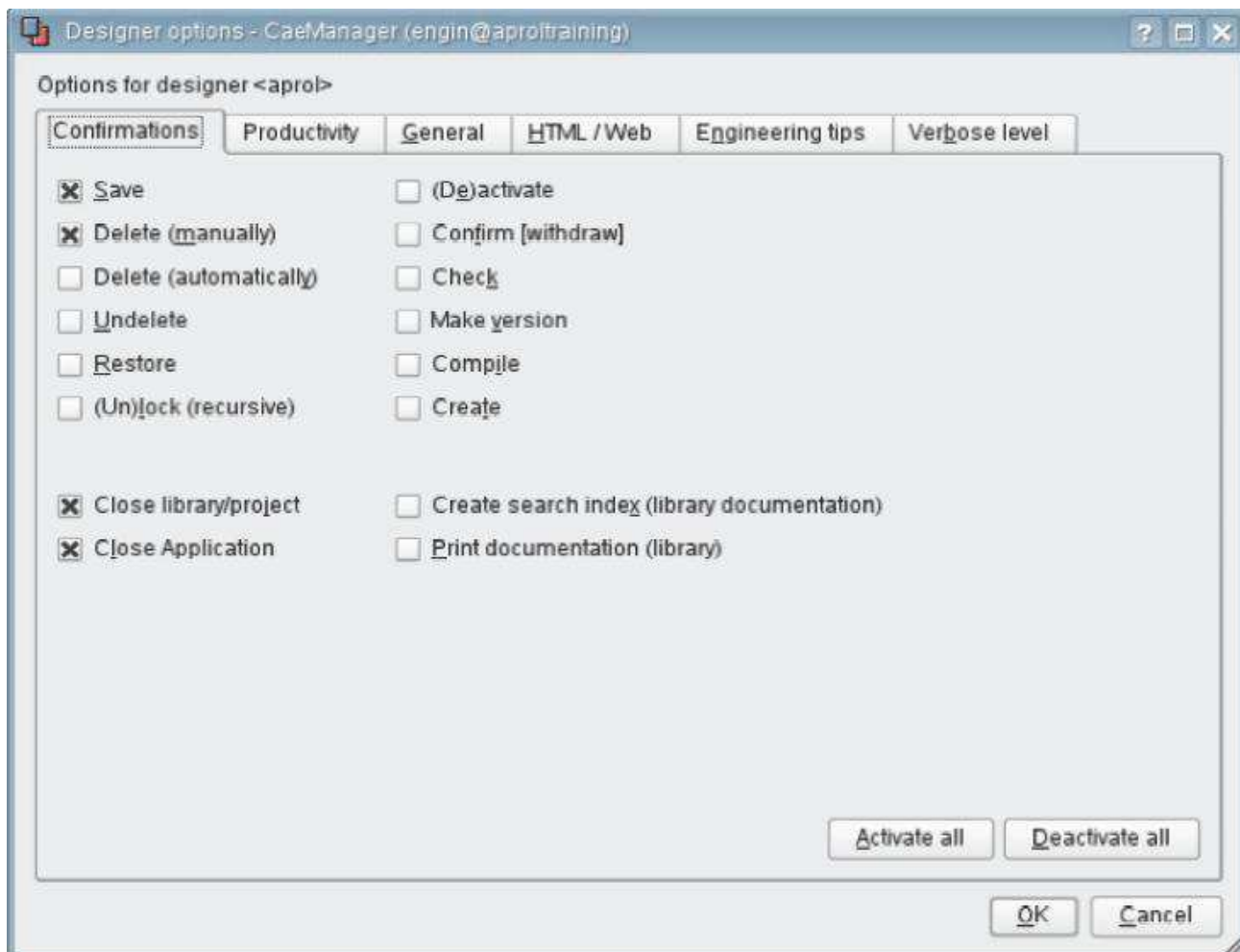


Рис. 6. Параметры разработчика

В зависимости от параметров вкладки **Productivity** (производительность) можно ускорить работу над разделами проекта за счет отсутствия необходимости выполнять дополнительные команды в SaeManager.

Установка параметра «**Check after saving**» (проверка после сохранения) запускает после сохранения промежуточную компиляцию, **результаты которой не сохраняются**. Можно использовать результаты данной компиляции для проверки правильности предыдущей конфигурации разработки. Вы не должны останавливать работу на текущей схеме!

Одновременно может задаваться только один из параметров: «Make version after saving» (создать версию после сохранения), «Activate after saving» (активировать после сохранения) и «Compile after saving» (компилировать после сохранения).

Если выбрать последний пункт меню, два предыдущих автоматически выполняются в фоновом режиме. Эти действия выполняются, т.к. требуются перед компиляцией.

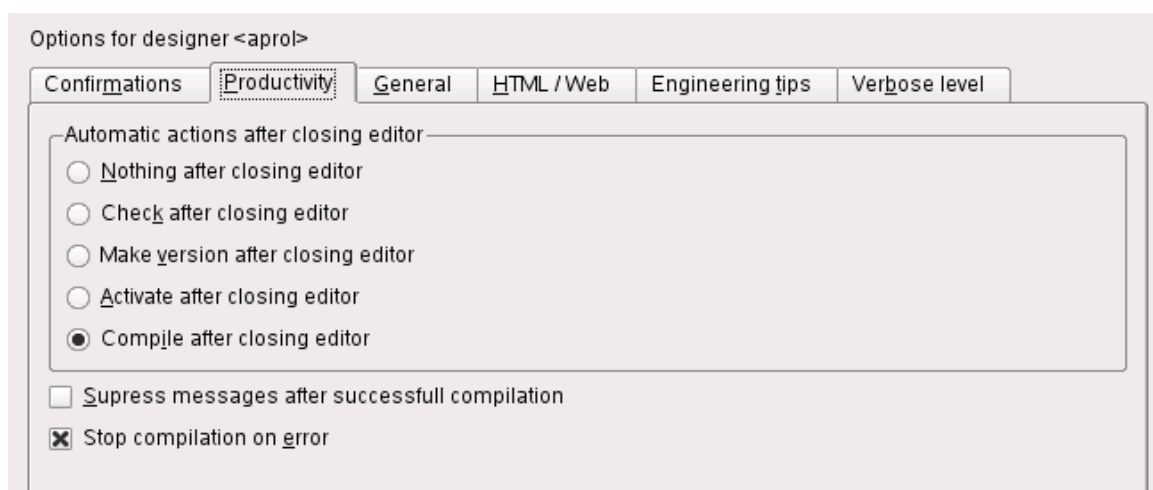


Рис. 7. Производительность

Разработчик также может выбрать предпочитаемый **HTML-браузер** и **HTML-редактор**.

В APROL включен ряд «**Engineering tips**» (советов по разработке). Их можно включать или отключать. Рекомендуется использовать данные советы на начальном этапе разработки проекта, т.к. в них содержится полезная информация.

Вкладка **Verbose Level** (уровень подробности) влияет на вывод информации при компиляции или генерации проекта. Чем выше выбранный уровень, тем более подробная информация выводится (verbose =

многословный, словоохотливый).

Примечание. Информация об ошибках

Сообщения об ошибках **не** подавляются уровнем подробности!



Рис. 8. Параметры сведений о компании

После выполнения основных настроек можно начинать собирать **команду проекта**.

Разработчикам, созданным в системе Designer Management, могут назначаться профили проекта при помощи пункта меню «**Extras / Project Profiles**» (дополнительные возможности / профили проекта). В данном пункте разработчики могут получить свой **определенный профиль проекта**.

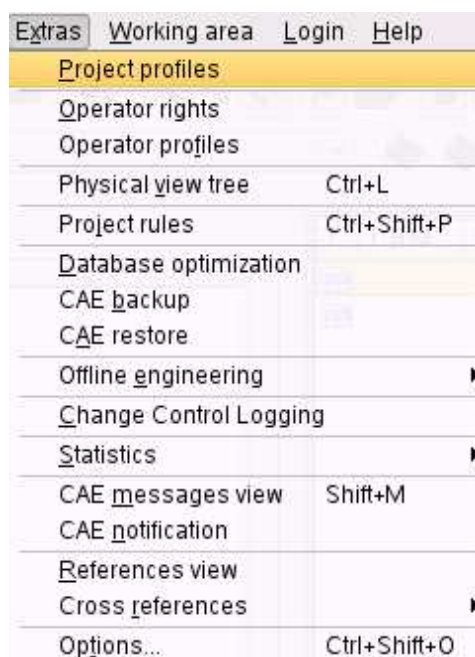


Рис. 9. Выбор профиля проекта

Может быть выбран один из **профилей проекта**, созданный в рабочей области Designer Management.

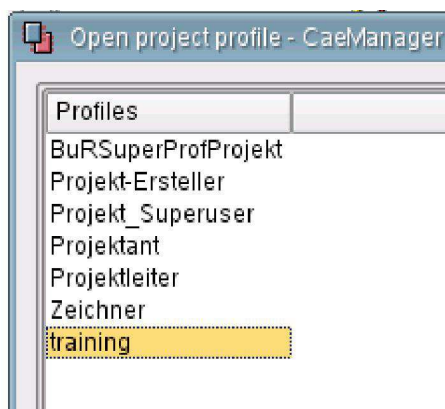


Рис. 10. Выбор профиля проекта

Теперь разработчики, работающие с данным **профилем проекта**, могут выбираться из списка всех разработчиков в диалоговом окне «**Assign project profile: <name of project profile>**» (назначение профиля проекта <название профиля проекта>). Назначение разработчику профиля и удаление одного или всех разработчиков профиля выполняется при помощи кнопок со стрелками.

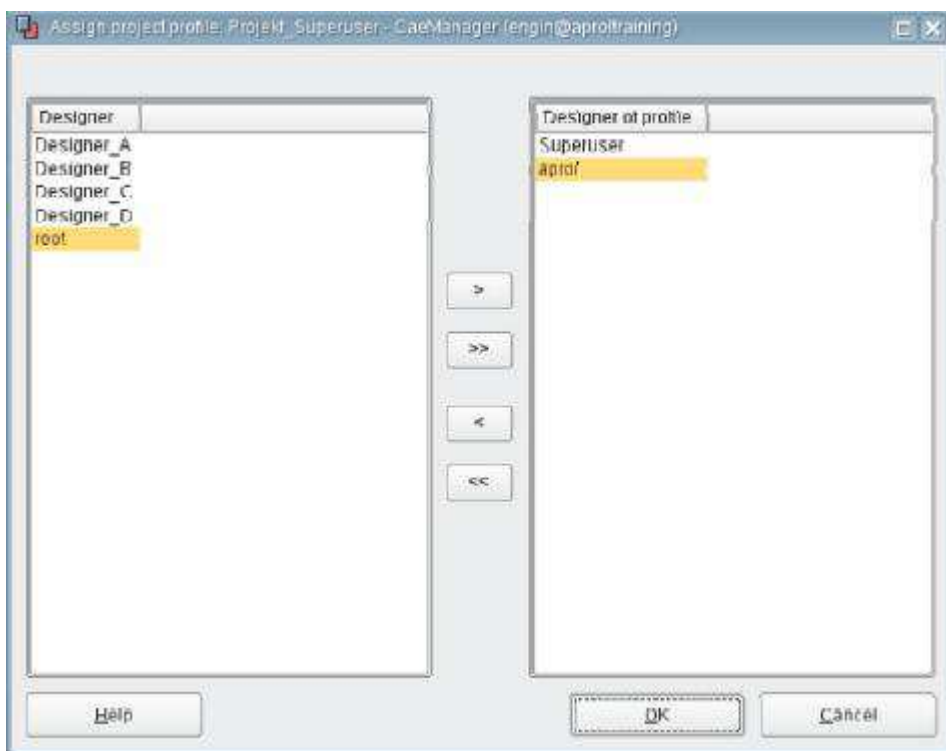


Рис. 11. Назначение профиля проекта разработчику

Данный способ управления позволяет легко организовать несколько проектов в одной базе данных. В результате один и тот же сотрудник одновременно может выступать в роли руководителя одного проекта и

иметь иные обязанности в другом (поддержка проекта и т.д.).

Структура прав проекта влияет на отдельные функции, необходимые для реализации системы. При этом упрощается планирование различных ресурсов (сотрудников) в соответствии с их образованием и квалификацией.

Примечание. Структура прав

В начале работы над проектом многие пользователи назначают права суперпользователя всем разработчикам, назначенным на проект. Иногда удобно по возможности быстро получить установленную структуру проекта без ограничения прав доступа. Однако при достижении определенного размера проекта (не позднее начала запуска в эксплуатацию) следует установить уровень управления.

Упражнение 1

В тренировочном проекте, созданном под вашим именем, назначьте разработчикам профили.

Место для подробного изложения:

4. ФИЗИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ

APROL предлагает возможность организации проекта в «географических» условиях. Это выполняется при помощи назначения каждому объекту аппаратуры (управляющему компьютеру, контроллеру) физического представления. Для этого сначала необходимо реализовать **Дерево физического представления**. Отображается конфигуратор, лежащий в основе проекта и служащий в качестве критерия реального местоположения, в котором устанавливается отдельный компонент управления процессом. Нет ничего необычного в том, чтобы система обозначений записей в дереве физического представления соответствовала реально существующим названиям уровней предприятия (название предприятия, название здания, помещения с электронным оборудованием, распределительного шкафа и т.д.).

Каждый сконфигурированный ЦП назначается своей ветке дерева физического представления. При создании приложения для определенной системы физическое представление назначается (в соответствии с деревом) созданным объектам (оборудованию, последовательности логических операций, диаграммам процессов). Так определяются пользовательские программы, которые необходимо создать для компонентов управления процессом. Совместно с объявлением оборудования дерево физического представления также определяет способ взаимодействия друг с другом отдельных компонентов процесса управления.

Уровни дерева физического представления создаются при помощи их технических идентификаторов и логического описания.

Внимание! Требования к СС (управляющему компьютеру)

В дереве физического представления должен находиться по крайней мере один управляющий компьютер и один ЦП для генерируемого проекта!

Внимание! Уровень контроллера

Управляющий компьютер должен располагаться по крайней мере на один уровень выше ЦП (контроллеров) в дереве физического представления.

Внимание! Ограничения

Т.к. сгенерированные пользовательские программы хранятся в структуре каталогов системы разработки под их физическими именами из дерева, название «Control Computer / Rack» (управляющий компьютер / стойка) (рядом с последним уровнем) должно быть уникальным и содержать не более 8 символов!

4.1. Создание дерева физического представления

В проекте пункт меню «**Extras / Physical view tree**» (дополнительные возможности / дерево физического представления) используется для открытия диалогового окна «**Physical view tree: <Name of project>**» (дерево физического представления <название проекта>).

После выбора уровня щелчком правой кнопки мыши открывается меню, в котором пользователь может создать «**New Level**» (новый уровень) под текущим или добавить к нему «**New**» (новый) элемент.

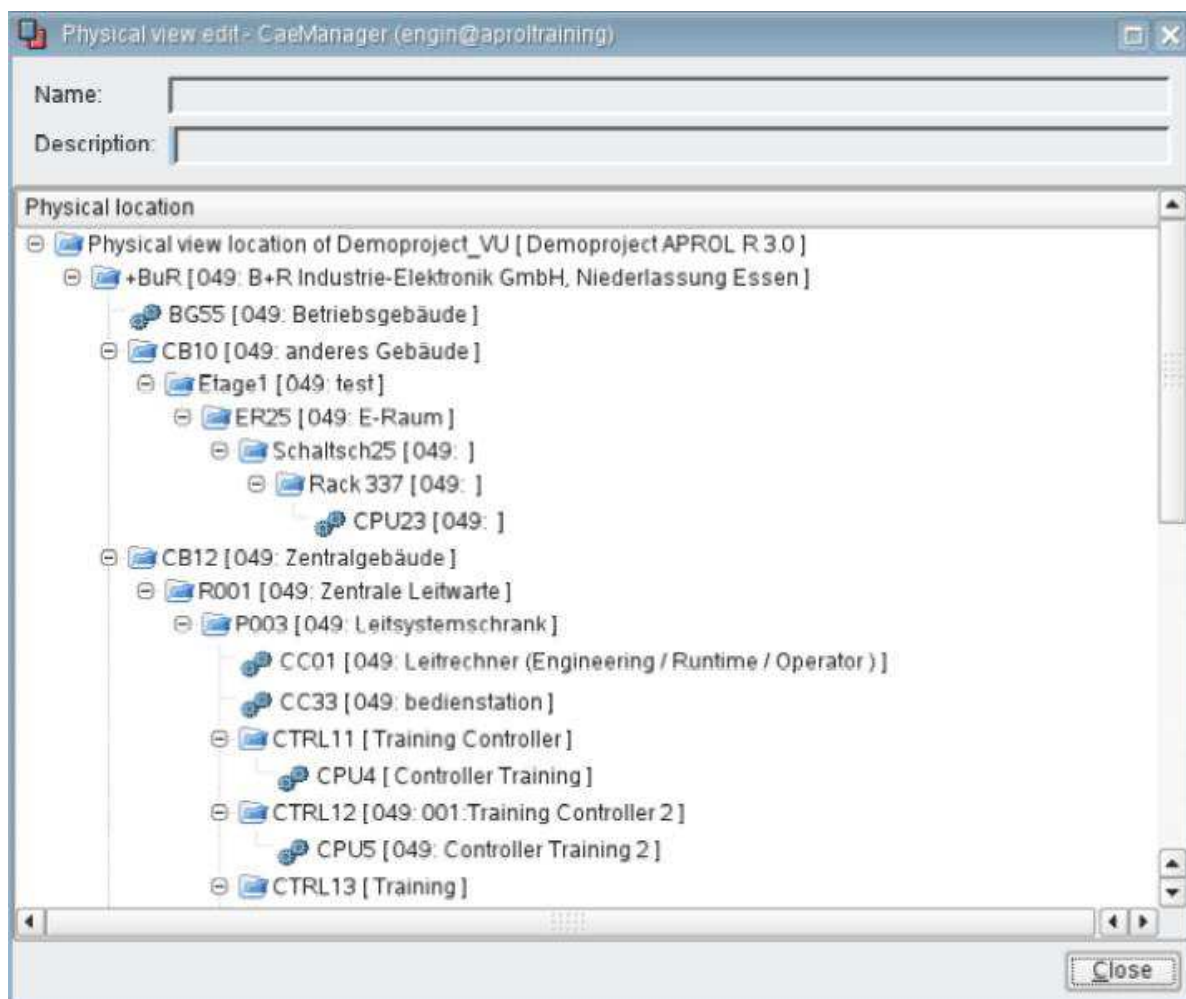


Рис. 12. Развертывание физического представления

Новый элемент или новый уровень вводится в следующем окне.



Рис. 13. Создание новой ветки в дереве физического представления

В дереве не существует ограничений на количество записей, поэтому проект можно в любое время расширить.

Примечание. Система обозначений

Необходимо помнить, что последний уровень управляющего компьютера и предпоследний уровень контроллера используются для генерации кода. Это означает, что синтаксис этих двух уровней должен соответствовать языку C.

Упражнение 2

Создайте физическое представления для «фиктивного» контроллера.

Место для подробного изложения:

5. СТРУКТУРИЗАЦИЯ ПРОЕКТА

Проект необходимо тщательно структурировать, чтобы сделать его наглядным и понятным для всех пользователей. **APROL** допускает вложение каталогов друг в друга, за счет чего упрощается структуризация проектов.

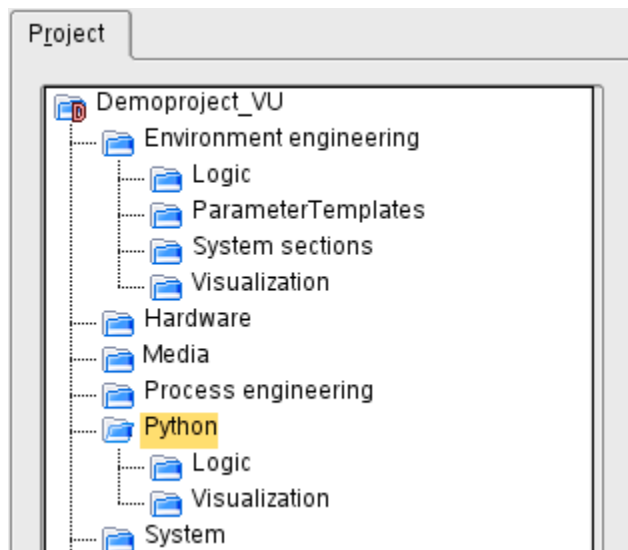


Рис. 14. Структура проекта

5.1 Вставка каталогов

В области навигации выберите каталог, в который требуется вставить другой подкаталог. Выберите пункт меню «**Create part / Directory**» (создать элемент/ каталог), с помощью которого открывается следующее диалоговое окно.

Рис. 15. Создание каталога

Create directory - CastManager (engin@aproltraining)

Directory:	Logic
Description:	area of secondary sedimentation basin
Name:	NAME
Description:	DESCRIPTION

OK Cancel

Как правило, в каталогах хранится соответствующая программная логика для аппаратуры контроллера. Задачи с именами, совпадающими или похожими на имена каталогов, создаются в области разработки аппаратного обеспечения.

Разумеется, в конечном счете соответствующий партнер по разработке решает, каким образом будет структурирован проект. В качестве

примера можно использовать демонстрационный проект. Помимо него, в распоряжении имеется руководство по разработке проектов.

6. КОНФИГУРАЦИЯ ОБОРУДОВАНИЯ

На данном этапе будет настраиваться контроллер. В рамках этого процесса в CaeManager будет определяться только количественная структура, т.к. состав уже был определен в начале. Для упрощения используйте «Import options» (импорт параметров).

6.1. Вставка контроллера

В выбранный каталог новый контроллер можно добавить при помощи пункта меню «**Create part / Controller**» (создать элемент/ контроллер). То же самое диалоговое окно можно открыть щелчком правой кнопки мыши в области настройки CaeManager.

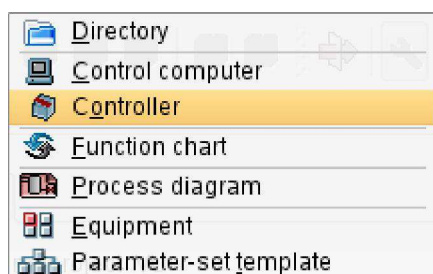


Рис. 16. Создание контроллера

В первой части следующего диалогового окна необходимо ввести уникальное имя и описание. Тип контроллера выбирается во второй части диалогового окна:

Main station

Контроллер с отдельным ЦП

Powerlink station

Стойка расширения, которая подключается к главной станции (Intel) по шине Powerlink

2005

Контроллер серии 2005

2003

Контроллер серии 2003, который используется только в качестве станции Powerlink в APROL

Наконец, для основной стойки главной станции или станции Powerlink должно быть указано уникальное имя (в поле **Name of basic rack**).

Create controller hardware - Caemanager (engin@aproltraining)

Directory: Logic

Description: area of secondary sedimentation basin

Name: NAME

Name of basis rack: NAME_OF_BASIS_RACK

Description: DESCRIPTION

System selection

☒ Basis rack (CPU module) ☐ System 2010

☐ Remote rack (SG3) ☒ System 2005

☐ Powerlink rack (I/O modules) ☐ System 2003

OK Cancel

Рис. 17. Создание нового контроллера

Примечание. Аппаратное обеспечение контроллера

Серия 2005 компании B&R, представленная во введении, является системой по умолчанию.

6.2. Настройка контроллера

После вставки контроллера его можно выбрать в дереве. Дважды щелкните кнопкой мыши значок контроллера в области разработки SaeManager (области настройки), чтобы открыть область разработки аппаратного обеспечения.

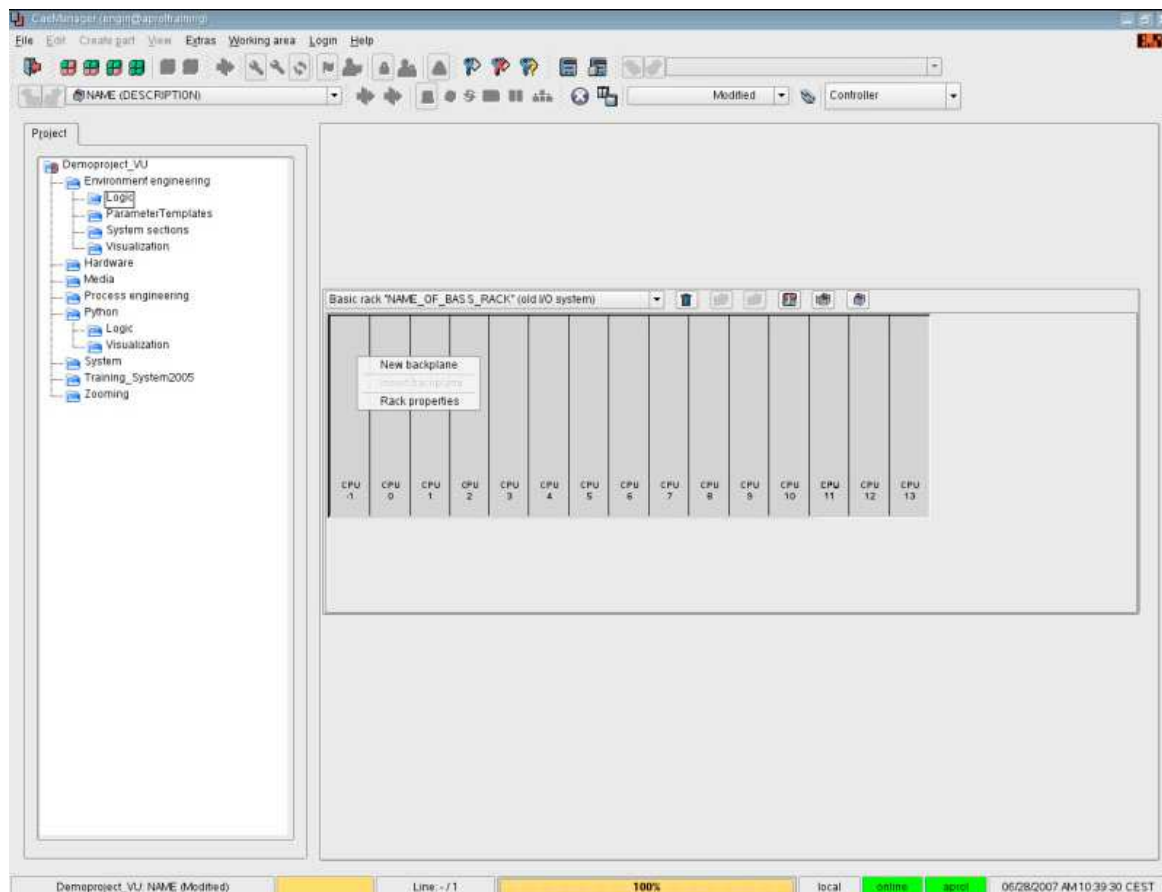


Рис. 18. Вставка объединительной платы при разработке аппаратного обеспечения

В области разработки под именем основной стойки появляется пустое конфигурационное пространство. Оно соответствует размеру стойки B&R-

2005. В этом пространстве контроллер может быть оснащен оборудованием и поэтапно настроен.

6.2.1. Вставка объединительной платы

При щелчке правой кнопкой **первого** слота появляется меню выбора. При выборе в меню пункта «**New backplane**» (новая объединительная плата) открывается окно «**Select backplane**» (выбор объединительной платы). Из списка можно выбрать подходящую объединительную плату.

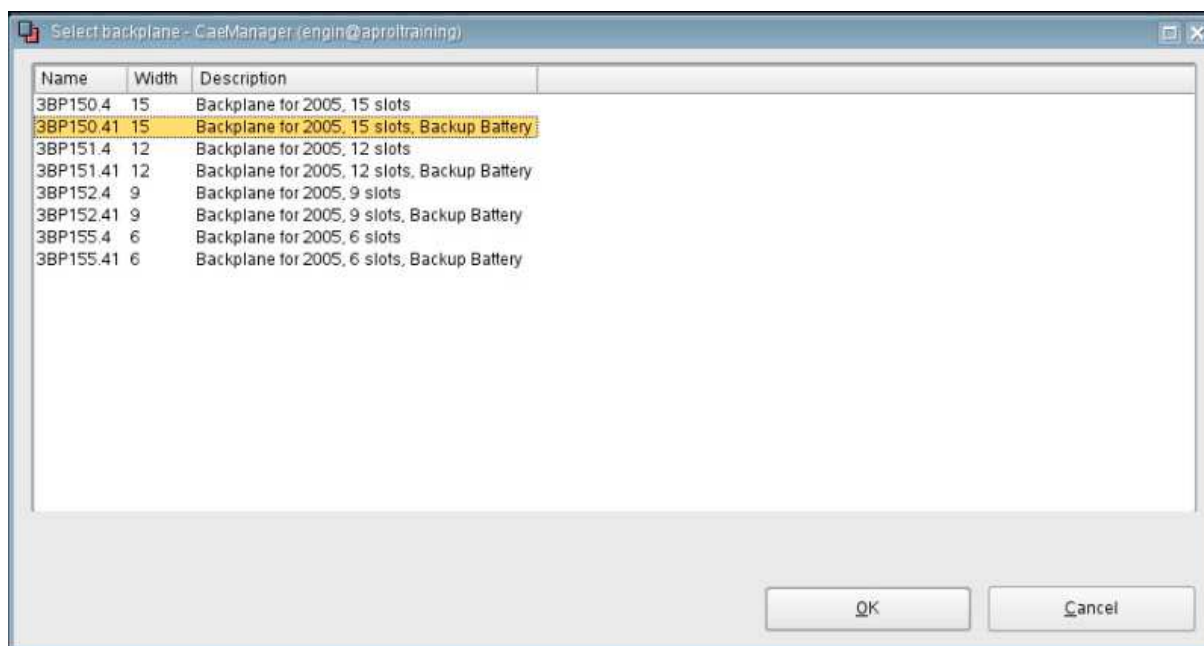


Рис. 19. Выбор объединительной платы ^^^

Примечание. Объединительная плата

Объединительная плата может помещаться только в самый дальний слот слева из-за адресации слота. Адресация хранится без изменений и может быть вызвана при выборе «**Backplane properties**» (свойства объединительной платы). Левая сторона всегда предназначается для

размещения модулей.

6.2.2. Размещение модулей

Теперь модули могут быть размещены на объединительной плате. Это выполняется точно так же, как и проектирование самой платы.

Щелкните правой кнопкой мыши слот, в который нужно добавить модуль. Выберите пункт меню «**New module**» (новый модуль). В последующем списке выбора можно указать соответствующий модуль и таким образом укомплектовать объединительную плату.

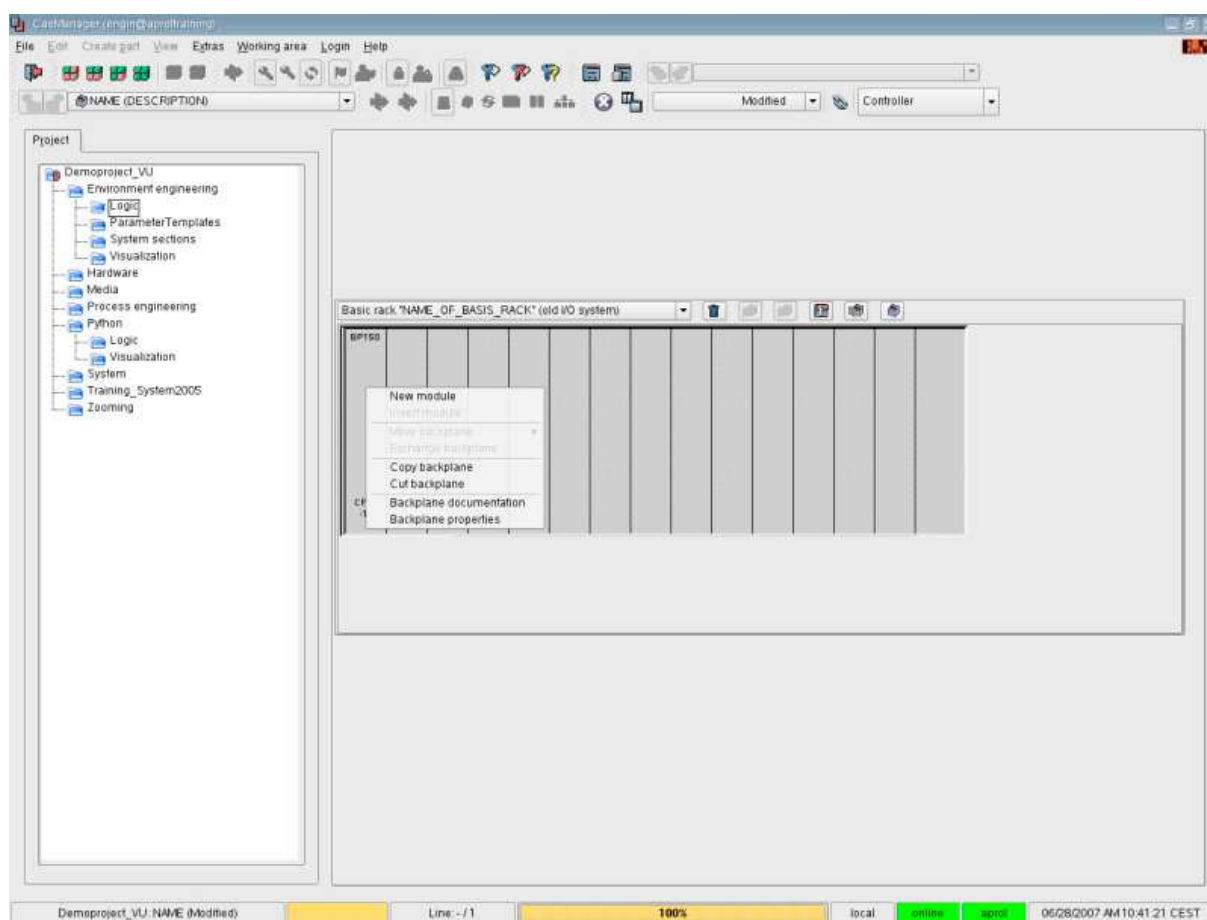


Рис. 20. Создание нового модуля (ЦП, I/O и т.д.)

При монтаже «логические ошибки» невозможны, потому что для выбора доступны только те модули, которые могут быть подключены в указанной позиции.

6.2.3. Документация модуля

Для получения дополнительных технических сведений в любой момент может быть вызвана документация, относящаяся к отдельному модулю. Справочная информация соответствует данным из руководства пользователя по аппаратному обеспечению B&R .

Сама документация предоставляется в виде HTML, содержание которого просматривается при помощи веб-браузера, настроенного для проектировщика. Для перехода на различные страницы документа могут использоваться ссылки.

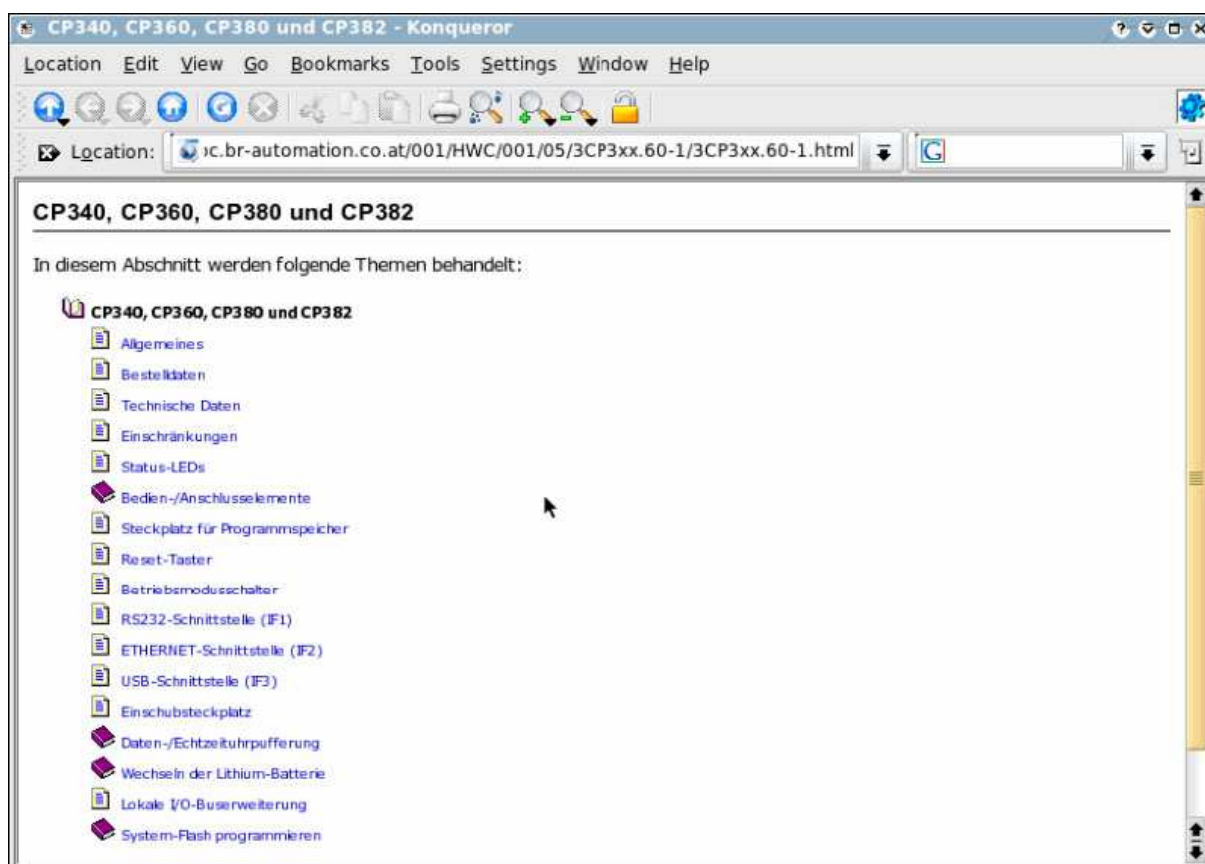


Рис. 21. Документация на аппаратное обеспечение

6.2.4. Свойства модуля

После размещения всех модулей их можно настроить. Свойства модуля можно настроить, щелкнув правой кнопкой мыши и выбрав пункт меню «**Module properties / I/O module**» (свойства модуль / модуль ввода / вывода), или просто дважды щелкнув модуль. В меню также имеется пункт «**Configuration**» (конфигурация). Его можно выбрать, только если уже выбран «новый» AR (>2.80) на ЦП. При помощи данного пункта меню

открывается окно с дополнительными возможностями настройки переменных системы, отслеживаемых модулей и ЦП.

6.2.5. Свойства ЦП

Все параметры, относящиеся к ЦП, задаются в диалоговом окне, открываемом при выборе пункта меню «**Module properties / CPU**» (свойства модуля/ ЦП) или двойным щелчком на ЦП.

- Назначение созданного ранее **физического представления**
- Создание **задач** различных классов
- Вставка **модулей V&R** (библиотек)
- Вставка **модулей проекта** (с модулями, настроенными в AS)
- Вставка **модулей исключений** (не допускается удаление при загрузке)
- Изменение операционной системы
- Установка **параметра загрузки** (настройка Ethernet для диспетчера загрузки)
- Установка свойств **Ethernet** (параметры для контроллера, компонент «sysconf»)
- Установка параметров операционной системы в «**sysconf**»
- Создание и заполнение **модулей данных**
- Настройка протоколов связи и коммуникационных интерфейсов (связь контроллер-контроллер и т.д.)
- Настройка подключений проекта (дополнительные подключения, подключения, учитывающие требования заказчика)

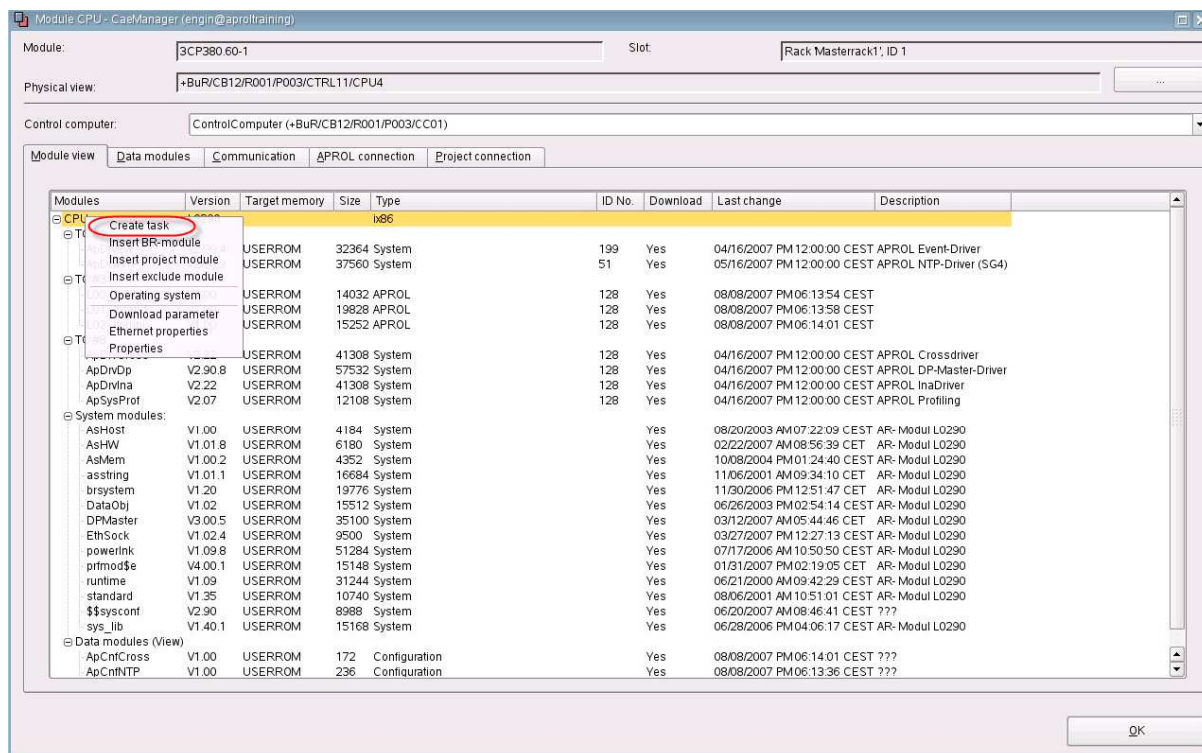


Рис. 22. Свойства модуля ЦП

6.2.6. Назначение физического представления

Назначение физического представления однозначно определяет ЦП, выполняющий управление в проекте. Это упрощает понимание проекта и позволяет техническому персоналу при помощи физического представления легко определять расположение соответствующего контроллера.

Кнопка «**Physical View**» (физическое представление) используется для открытия дерева физического представления. По дереву можно перемещаться и выбирать физическое представление. Выбор осуществляется двойным щелчком или нажатием «**OK**».

Это подтверждение возможно только в случае нахождения на последнем уровне созданного физического представления.

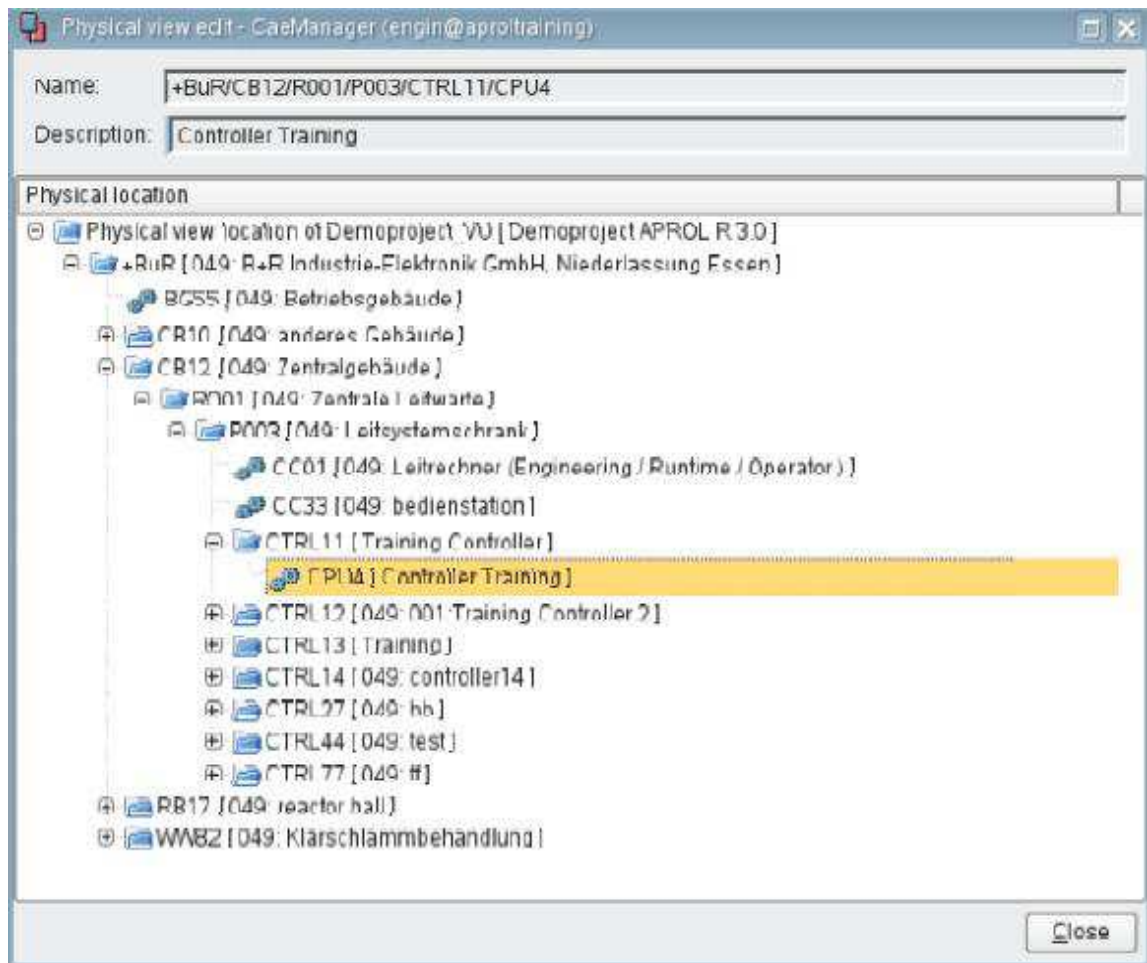


Рис. 23. Назначение физического представления

Соответствующий управляющий компьютер также может быть назначен в записи «**Control computer**». Это необходимо для многозадачного выполнения на управляющем компьютере (определяются несколько задач управляющего компьютера).

6.2.7. Создание задачи

Новая задача может быть создана щелчком правой кнопки мыши ЦП или класса задач в представлении модуля. Ее название должно соответствовать структуре проекта, а описание – давать представление о выполняемых функциях. Можно создавать задачи всех имеющихся классов, общее

количество задач ограничивается только объемом имеющейся на ЦП памяти.

Вновь созданная задача может назначаться в других структурных схемах разработки. В результате эти структурные схемы выполняются данной задачей.

Примечание. Задача

Количество задач должно точно соответствовать потребностям, т.к. каждая задача требует места на контроллере для хранения файловой информации и времени на вычисления (планировщик задач). Для каждой новой функции не имеет смысла заводить отдельную задачу. Функции с четкими взаимосвязями следует компилировать в рамках одной задачи и обрабатывать вместе.

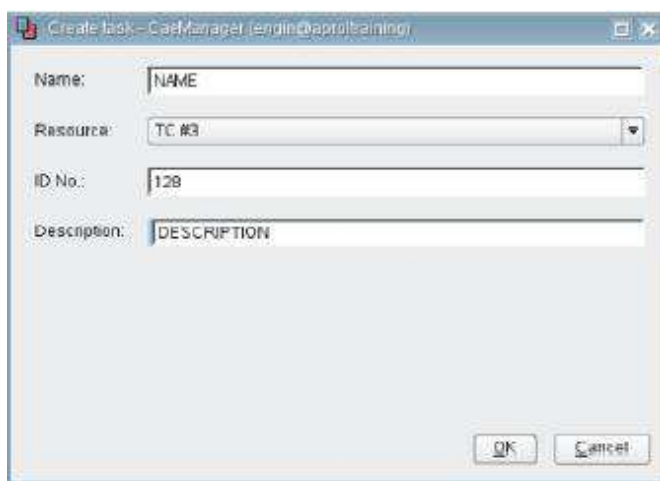


Рис. 24. Создание задачи

6.2.8. Вставка модулей

Модули с расширением «*.br» могут вставляться при помощи «**Controller hardware engineering**» (проектирование аппаратного обеспечения контроллера). Существует два типа модулей – модули BR и модули проекта. Модулями BR являются библиотеки и функции, поставляемые непосредственно *APROL*. (Обычно это библиотеки Automation Studio.)

Модули проекта – это файлы с расширением «*.br», разработанные непосредственно для проекта. В большинстве случаев Automation Studio является средой разработки, используемой вместе с этими модулями.

6.2.9. Исклучение модулей

Данная функция необходима для недопущения удаления модулей. Контроллер B&R динамически создает несколько модулей (интерактивно на ЦП), которые не располагаются в списке загрузки. Это означает, что по умолчанию эти модули удаляются во время загрузки, чего обычно не требуется. Поэтому данная функция используется для указания модулей, которые не следует удалять при загрузке.

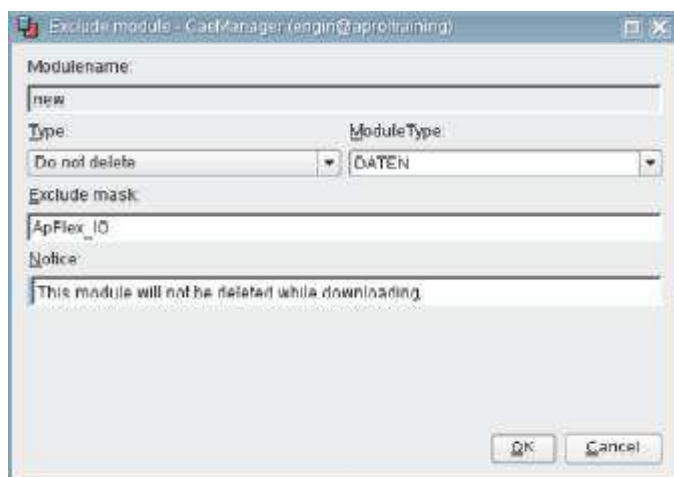


Рис. 25. Модули исключения

Также предоставляется функция «**Do not change**» (не изменять). При использовании данного параметра введенный модуль никогда не будет изменен.

Параметр «**ModuleType**» (тип модуля) определяет тип модуля.

Можно использовать сокращения (например, Ap*) или вводить целиком имена файлов непосредственно в поле «**Exclude mask**» (шаблон исключения).

6.2.10. Выбор операционной системы

В *APROL* для совместимости постоянно расширяется количество представляемых операционных систем. Помимо этого, в любое время может быть импортировано ПО AR, предоставляемое B&R.

Выбор пункта меню «**File / OS Manager**» (файл/ диспетчер ОС) или щелчок правой кнопки мыши по объекту ЦП открывают всплывающее окно из меню «**Operating system**» (операционная система). После выбора появляется следующий запрос:

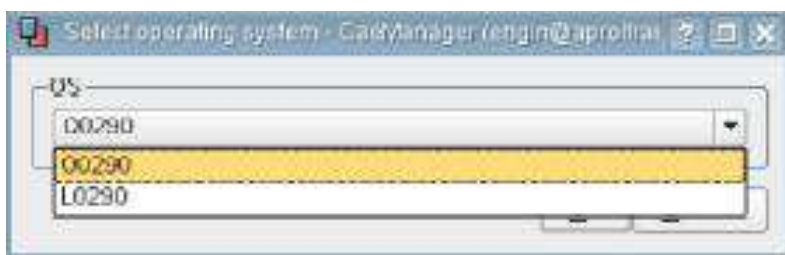


Рис. 26. Выбор требуемой операционной системы

После выбора операционной системы подтвердите его при помощи «**ОК**».

6.2.11. Параметры загрузки

Задача загружается при помощи другого приложения, называемого DownloadManager. Для этого приложения требуются данные, определяющие способ загрузки. Для задания всех необходимых параметров используются следующие шаблоны.

Рис. 27. Параметры загрузки

Download parameter - CaeManager (engin@aproltraining)

Base configuration

Medium: INA-Service Ethernet Download mode: copyMode

Route: setRT: 3000

☐ Autostart ☐ noDialog

Medium specific parameters

IP address configuration

IP address: 192.168.76.31 MyNode: 231

OK Cancel

Один из наиболее важных параметров – «**Download mode**» (режим загрузки). Мягкая загрузка возможна только при использовании «**copyMode**». В данном контексте мягкая загрузка означает, что ввод / вывод не будет отключаться, тем самым допуская «оперативное» расширение системы в любое время.

6.2.12. Свойства Ethernet

Свойства Ethernet являются составной частью конфигурации системы ЦП B&R (**sysconf**). С этой целью создан интерфейс, т.к. данные свойства обычно требуют настройки. Ниже на рисунке показаны общие параметры. Подробные сведения могут быть получены из справки, открываемой при нажатии кнопки «**Help**».

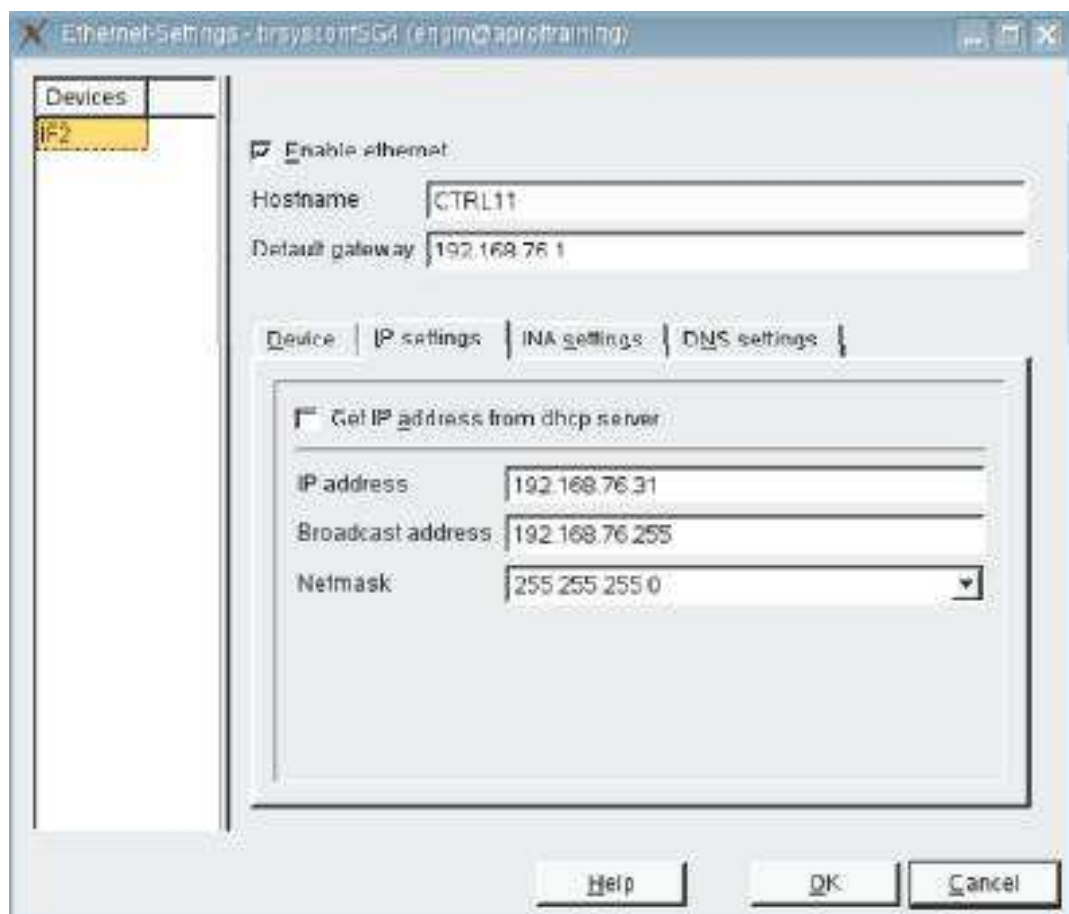


Рис. 28. Параметры Ethernet

Устанавливаемые параметры должны соответствовать общим правилам сети Ethernet.

6.2.13. Свойства (sysconf)

Параметры системы ЦП (**sysconf**) настраиваются при помощи пункта меню «**Properties**» (свойства). В параметры включаются, например, настройки классов индивидуальных задач, взаимосвязи или управление языком. Для получения подробной информации см. Документацию *APROL*.

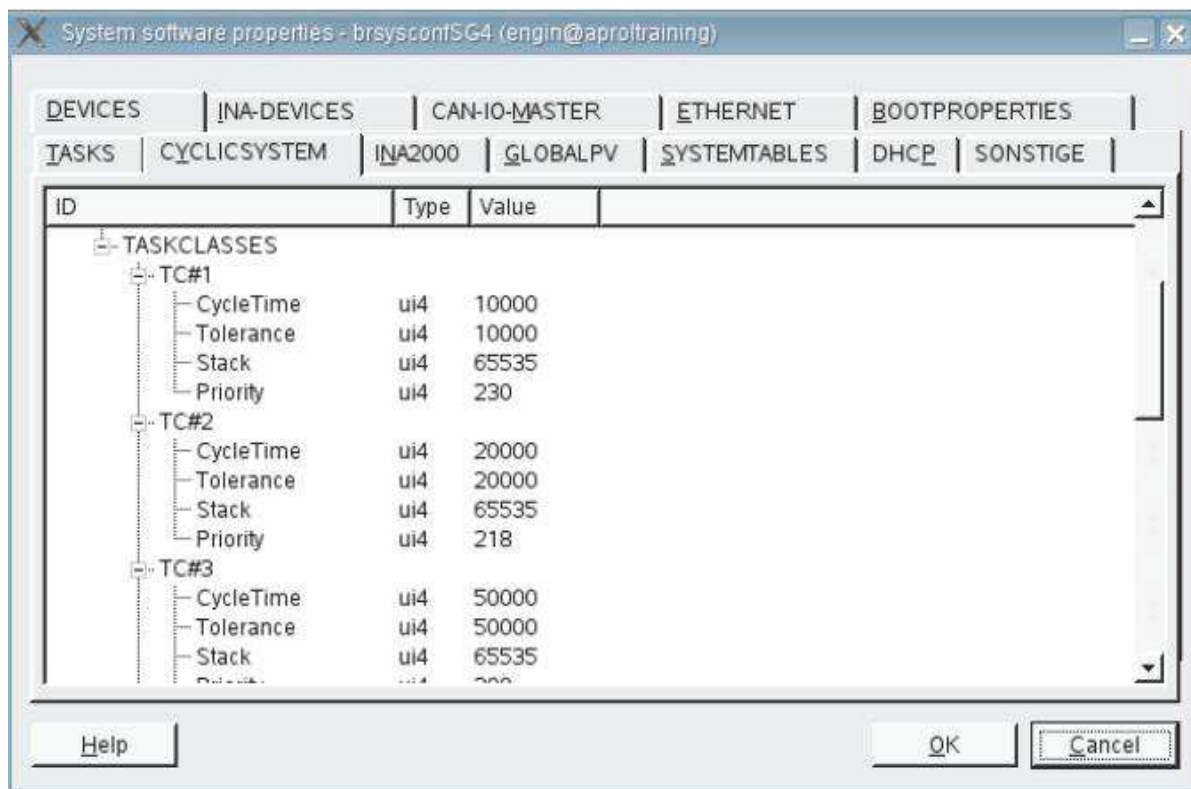


Рис. 29. Параметры Sysconf (конфигурация системы ЦП)

6.2.14. Модули данных

Для настройки или обработки модуля данных часто требуются некоторые функциональные возможности.

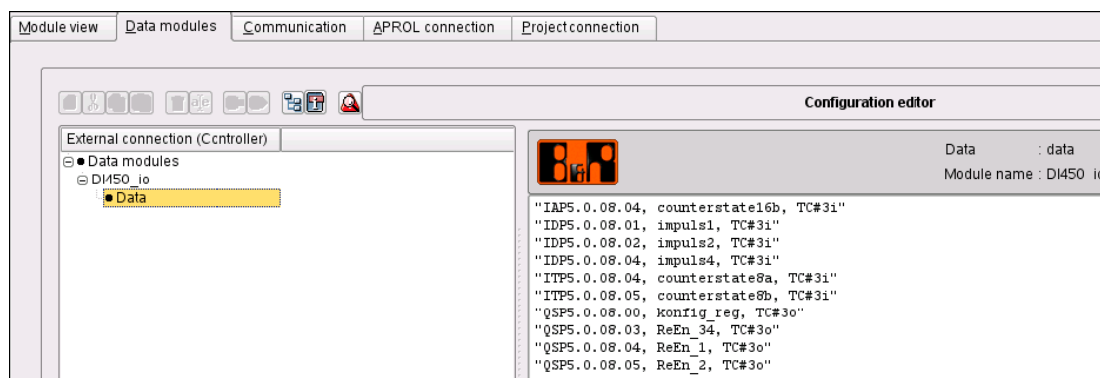


Рис. 30. Модуль данных

Точно так же, как в AS (Automation Studio), модуль данных создается в разделе «**Data modules**» (модули данных). На снимке экрана приведен пример использования платы счетчика.

6.2.15. Связь

Возможности стандартного определения из APROL предоставляются в области «**Communication**» (связь). Основной конфигурацией является подключение вида контроллер-контроллер.

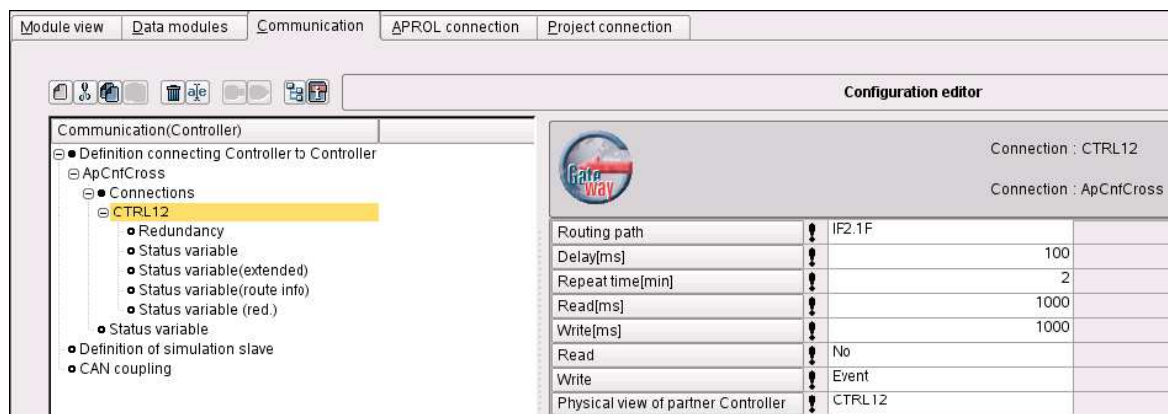


Рис. 31. Подключение контроллер-контроллер

За счет непосредственной связи между контроллерами упрощается обмен данными (в результате не требуется передачи переменных через управляющий компьютер (InaDriver)). По существу пользователь задает канал связи (маршрутизацию). Система автоматически определяет, следует ли «захватить» переменную, т.к. передача переменной осуществляется по физическому представлению.

Поэтому, если в структурной схеме контроллера А на разъем записываются данные, которые считываются в структурной схеме контроллера В, то для переменной автоматически определяется тип подключения контроллер-контроллер. Более того, пользователь может также определить, возможна ли связь в обоих направлениях (чтение, запись) и, в случае записи переменных, происходит ли она циклически или синхронизируется по событиям. Часто настраивается только СОБЫТИЕ записи (контроллер А) и СОБЫТИЕ чтения на парном контроллере (контроллер В).

Примечание. Подключение контроллер-контроллер

Настоятельно рекомендуется организовывать непосредственную связь между контроллерами, т.к. при этом увеличивается надежность. Управляющий компьютер никогда не обладает таким же параметрами надежности, как контроллер.

Подробные сведения о данном и других типах связи см. в документации *APROL* и курсе обучения для высококвалифицированных специалистов (модуль TM850).

6.2.16. Подключения APROL

Область подключений APROL содержит стандартные подключения к внешним устройствам. Также реализовано подключение Profibus DP. Каждое подключение требует различной настройки. Отдельные подключения могут быть изучены на семинарах или реализованы с помощью интерактивной справочной системы *APROL*.

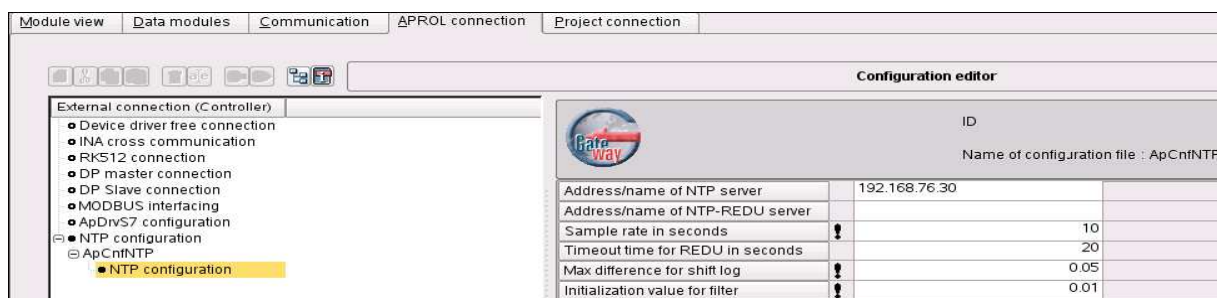


Рис. 32. Пример конфигурации NTP

6.2.17. Подключения проекта

В данной области может быть выполнено пользовательское подключение. Для этого сначала необходимо определить соответствующее правило при помощи редактора правил проекта. Для получения подробных сведений см. справочные файлы *APROL*.

6.2.18. Свойства модулей ввода / вывода

При двойном щелчке мыши модуля или выборе пункта из его контекстного меню (так же, как и для ЦП) открывается диалоговое окно «**I/O module configuration**» (настройка модулей ввода / вывода). Все аппаратные устройства ввода / вывода, представленные пользователям проекта для выбора, настраиваются в полях окна.

Примечание. Устройства ввода / вывода

В дальнейшем неправильный ввод при разработке проекта не допускается, т.к. устройства ввода / вывода вводятся однократно, а затем выбираются из списка.

IO module configuration - CaeManager (engin@aproltraining)

Module: 3AI350.6 Slot: Rack Masterrack1 ID 6

Channel	Signal Mode	Sign.Bit	I	E	Name	MSR No.	IEC type	Substitute value	Resulting substitute	A	AOS	MR begin	MR end	Unit	Description
1	ModuleOk	Status	☐	☐			BOOL			☐	☐			---	Modul Status (1 = Modul inserted)
2	AnalogInput01	0V..+10V	12/12	☐	PT01	PT-01	REAL			0	☐	0	190	kg/cm2	analog Input
3	AnalogInput02	0V..+10V	12/12	☐	TT01	TT-01	REAL			0	☐	0	100	°C	analog Input
4	AnalogInput03	0V..+10V	12/12	☐	TT04	TT-04	REAL			0	☐	0	100	°C	analog Input
5	AnalogInput04	0V..+10V	12/12	☐	SC08	SC-08	REAL			0	☐	0	100	%	analog Input
6	AnalogInput05	0V..+10V	12/12	☐			REAL			☐	☐				±10 V, Resolution 12 Bit
7	AnalogInput06	0V..+10V	12/12	☐			REAL			☐	☐				±10 V, Resolution 12 Bit
8	AnalogInput07	0V..+10V	12/12	☐			REAL			☐	☐				±10 V, Resolution 12 Bit
9	AnalogInput08	0V..+10V	12/12	☐			REAL			☐	☐				±10 V, Resolution 12 Bit
10	ModuleError	Status	☐	☐			BOOL			☐	☐			---	Module error (1 = Error)
11	Averaging	Status	☐	☐			BOOL			☐	☐			---	Averaging (1 = aktiv)

Import Export OK Cancel

Рис. 33. Настройка устройств ввода / вывода

Вся информация, влияющая на ввод / вывод, может быть настроена (если это допускается). Настройки включают тип, тип сигнала, сигнальный бит, переменную события, имя, номер MSR, тип данных, начало диапазона измерений, конец диапазона измерений, единицы измерений и описание.

Эта информация предоставляется для разработки структурной схемы, и ее следует использовать. Преимущество данного метода заключается в том, что данными можно управлять из одной точки.

Примечание. Масштабирование канала

Границы диапазона измерений определяются централизованно, и значение коэффициента масштабирования устанавливается для проекта в целом. Если в ходе ввода в эксплуатацию обнаруживается, что для кодирующего устройства отсутствует заранее определенный диапазон измерений, то соответствующий диапазон может быть быстро применен без поиска функционального блока масштабирования в структурной схеме.

Назначение устройств ввода / вывода может выполняться для каждого модуля при помощи импорта файла из систем планирования.

Модуль ввода / вывода может быть настроен из пункта меню «**Module properties / Configuration**» (свойства модуля / настройка).

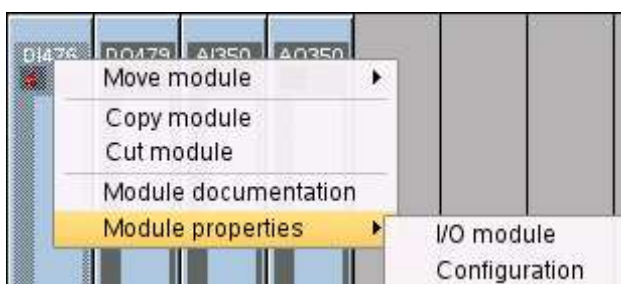


Рис. 34. Вызов настройки модуля ввода / вывода

Конечно, данные для каждого модуля различаются. Технические данные см. в справочных файлах к соответствующему модулю или в интерактивной справке *APROL*

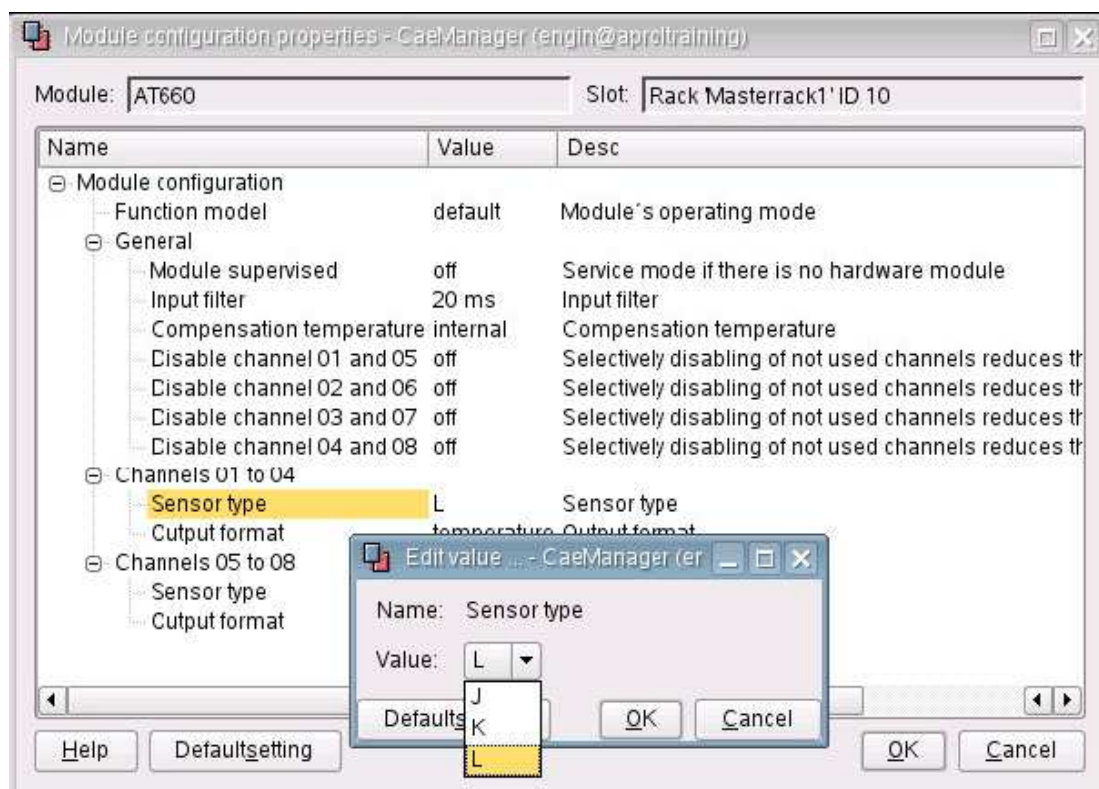


Рис. 35. Пример настройки модуля AT660

На снимке экрана приведен пример настройки. Конфигурационные возможности в основном не требуют объяснений.

6.2.19. Сохранение изменений

После завершения настройки изменения сохраняются при помощи пункта меню «**File / Save**» (файл/ сохранить). При этом, а также при других изменениях, генерируется и сохраняется новая версия. Система отображает полную информацию о версии в диалоговом окне. Помимо этого, пользователь может добавить **комментарий** к этой информации. Кнопка «**New Release**» (новый выпуск) позволяет увеличить на единицу номер выпуска.

Примечание. Управление версиями

Управление версиями не регулируется. Номер версии задается объекту один раз.

Save as - CdmManager (engine@aproltraining)

General

Name: NAME

Description: B&R Controller for APROL Training

Created: 06/28/2007 PM 12:16:14 CEST aprol

Edited version

Designer: 06/28/2007 PM 12:16:15 CEST aprol

Version: Modified

Saving

Designer: aprol

Version: 1 0

New Release

Version comment:

Ethernet settings changed

OK Cancel

6.2.20. Активация и компиляция

Отдельные версии объекта можно сохранить в базе данных разработки

APROL

Чтобы указать системе, какие из сохраненных версий следует использовать для генерации и последующего использования проекта, определенная версия **активируется**. Активация версии должна производиться в представлении версий объекта в области разработки SaeManager. Перейти в представление версий можно, выбрав пункт меню «**Versions**» (версии) в контекстном меню объекта (открывающемся при щелчке правой кнопки мыши). Если требуется активировать последнюю сохраненную версию, выберите пункт контекстного меню объекта «**Active version**» (активная версия).

«**Compile**» (компиляция) проверяет возможность **генерации** объекта. Любые сделанные настройки объекта проверяются на достоверность и правильность синтаксиса.

	Copy	Ctrl+C
	Paste	Ctrl+V
	Rename	
	Delete	Ctrl+D
	Check	F5
	Create version	F6
	Activate	F7
	Compile	F8
	Check (active version)	Shift+F5
	Compile (active version)	Shift+F8
	Generate	F9
	Deactivate	Shift+F7
	Compile/Generate all	Shift+F9
	Set bookmark	F11
	Remove bookmark	Shift+F11
	Lock	Shift+L
	Unlock	Shift+U
	Unlock foreign	Ctrl+Shift+U
	Change Control Logging	Ctrl+L
	References	Shift+F12
	CAE messages	Ctrl+M
	Remove CAE advices	Ctrl+H
	Version view	Shift+V
	Properties	F12

Примечание. Производительность

В большинстве случаев меню «**Extras/Options...**» (дополнительные возможности / параметры) используется для настройки поведения, поэтому после сохранения версия автоматически активируется и компилируется.

6.2.21. Создание страниц комментариев

Для созданных объектов пользователи могут вставлять страницы комментариев. При этом предоставляются подробное описание объектов и документация по разработке. Страницы с комментариями располагаются в том же каталоге, что и описываемые в них объекты, что позволяет напрямую их согласовывать.

Список выбора в области меню SaeManager используется для переключения между страницами конфигурации аппаратного обеспечения и комментариями. На представлении проводника размещается значок, определяющий наличие страницы с комментариями для объекта.

T	L B M	State	Name	C	Instance	Description
		 	CTRL01_RH	---	---	B&R Controller reactor hall
		 	CTRL02_FT	---	---	B&R Controller digestion tower
		 	CTRL11_Training		---	B&R Controller for APROL Training

Рис. 38. Страница с комментариями

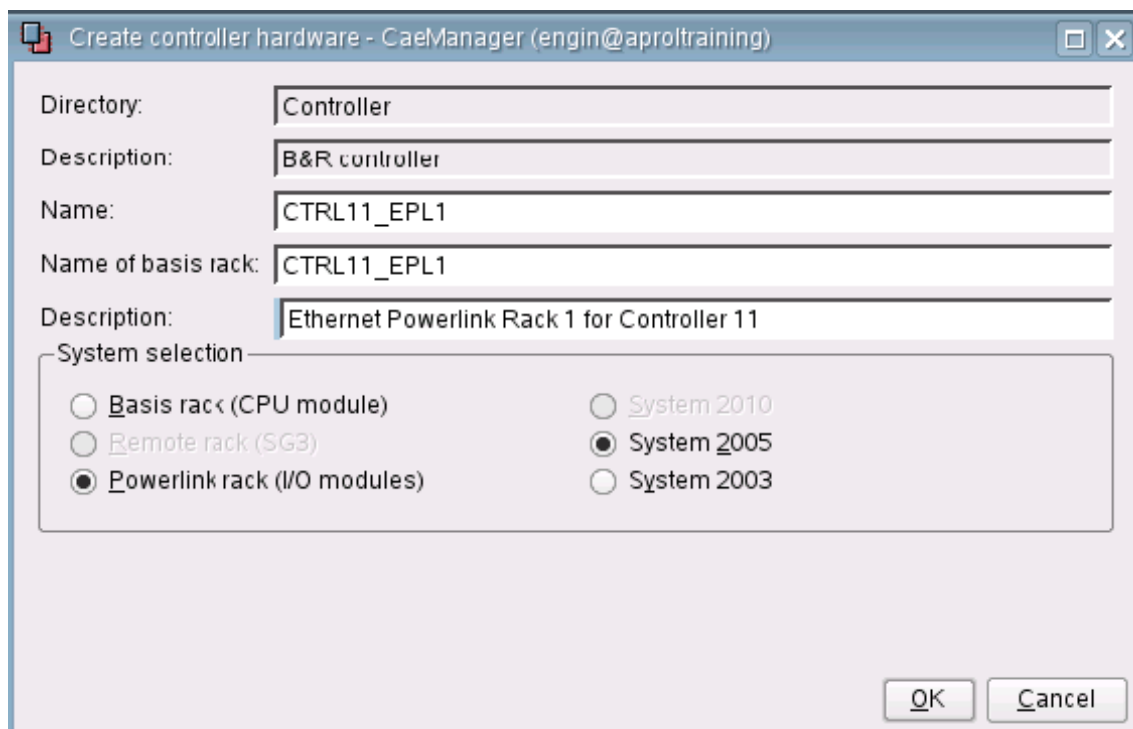
Изменения на странице с комментариями подпадают под действие управления версиями точно так же, как и изменения соответствующего объекта.

6.3. Настройка стойки Ethernet POWERLINK

В большинстве случаев в состав контроллера входит не только основная стойка. Обычно подключаются еще и стойки Powerlink. Это удаленные устройства ввода / вывода, подключенные по Ethernet (кроссоверный кабель кат. 5) Концентратор проводных соединений с топологией «звезда» или шинное соединение, или, при возможности, сочетание обоих методов. При расстояниях, превышающих 100 м,

следует использовать волоконную оптику с медиаконвертером. Подробные сведения об Ethernet POWERLINK см. в каталоге продуктов *APROL* и интерактивных справочных файлах *APROLL*.

Расширение POWERLINK создается так же, как и контроллер. В следующем входном трафарете следует выбрать параметр «**Powerlink rack**» (стойка Powerlink) вместо «**Main rack**» (основная стойка).



Create controller hardware - CaeManager (engin@aproltraining)

Directory: Controller

Description: B&R controller

Name: CTRL11_EPL1

Name of basis rack: CTRL11_EPL1

Description: Ethernet Powerlink Rack 1 for Controller 11

System selection

☐ Basis rack (CPU module) ☐ System 2010

☐ Remote rack (SG3) ☒ System 2005

☒ Powerlink rack (I/O modules) ☐ System 2003

OK Cancel

Рис. 39. Создание стойки EPL

Для стойки Ethernet POWERLINK пользователь может выбрать серию B&R 2005 или серию B&R 2003.

И опять проектирование аппаратного обеспечения выполняется при помощи размещения объединительных плат, источника питания, вставки резервного POWERLINK и модулей ввода / вывода.

При необходимости пользователь может переключиться на новую систему ввода / вывода, выбирая свойства стойки. Это обязательно при использовании на главной стойке версии AR старше, чем V280.

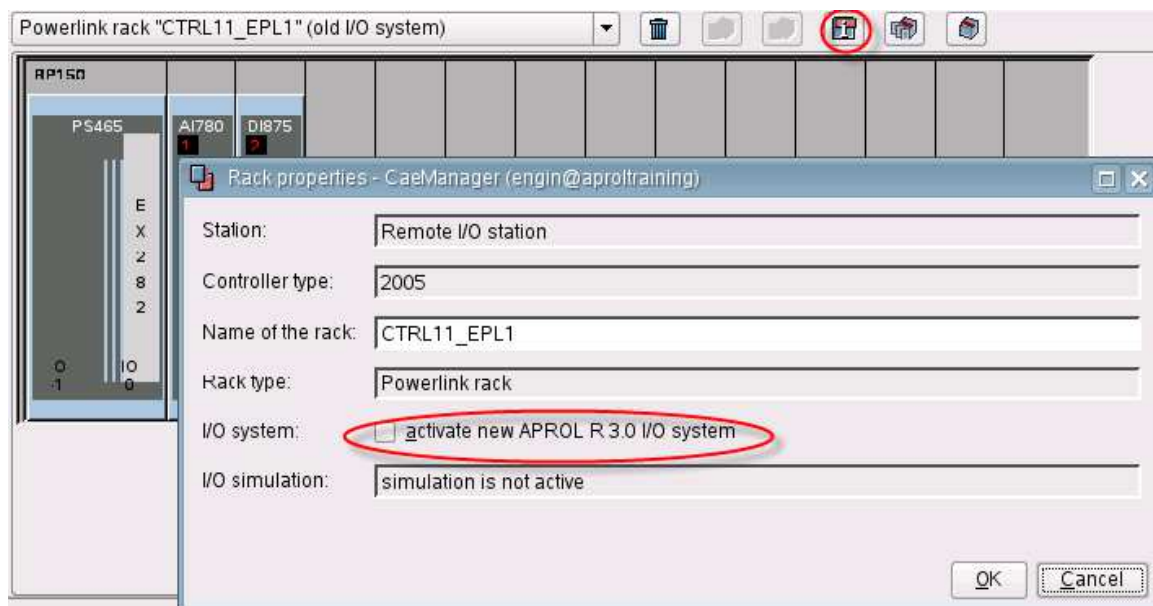


Рис. 40. EPL, новая система ввода / вывода

После завершения настройки стойки EPL конфигурация сохраняется, назначается версия, которая активируется и компилируется.

Затем необходимо установить связь с соответствующей основной стойкой. Для этого стойка должна быть открыта и к ЦП (подмодуль) должен быть добавлен главный модуль aPCI EPL. В следующем шаблоне может быть выбрана стойка EPL и определен и добавлен идентификатор подчиненного устройства (переключатель узла на подчиненном устройстве EPL для уникальной адресации).

Рис. 41. Подключение стойки EPL к ЦП

Define Powerlink master - CaelManager (engin@aproltraining)

Module: 3IF786.9 Slot: Rack Masterrack1, ID 1.1

Powerlink slave	ID	used
CTRL11_EPL1	1f	x

Master ID: 0x01

Slave No.:

Rack name: CTRL11_EPL1 ...

Slave ID: 1f

used: ☒

<= Add

Упражнение 3

Создайте фиктивный контроллер и измените один из ранее созданных. При этом логически структурируйте данную часть проекта при помощи каталогов и задач.

Место для подробного изложения:

7. КОНФИГУРАЦИЯ УПРАВЛЯЮЩЕГО КОМПЬЮТЕРА

Помимо только что сконфигурированного контроллера, в систему управления процессами *APROL* также входят компоненты управления

Как правило, компонентами системы управления процессами являются управляющие компьютеры. Как уже упоминалось во введении, системы разработки, системы для работы операторов и исполняющие системы могут сочетаться произвольным образом. Настройки компьютера, установленные для подобных комбинаций, предлагаются в **системном каталоге** *APROL*.

7.1. Добавление управляющего компьютера

Новый компонент системы управления процессами может быть вставлен в выбранный каталог при помощи пункта меню «**Create part / Control computer**» (создать элемент / управляющий компьютер).

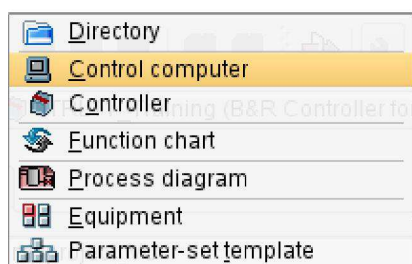


Рис. 42. Создание управляющего компьютера или станции оператора

В соответствующем диалоговом окне необходимо ввести уникальное имя и описание.



Рис. 43. Создание управляющего компьютера или станции оператора

7.2. Конфигурация управляющего компьютера

После вставки управляющего компьютера его можно выбрать в дереве. Дважды щелкните кнопкой мыши по значку управляющего компьютера в области разработки SaеManager (области настройки), чтобы открыть область разработки аппаратного обеспечения.

В следующем диалоговом окне вводятся все необходимые параметры управляющего компьютера.

The screenshot shows a configuration window titled 'Network' with several tabs. The 'Network' tab is selected. Inside, there are checkboxes for various server roles: 'Use as logging server' (checked), 'Use as documentation server' (checked), 'Use as NTP server' (unchecked), 'Set as default' (checked), and 'Set as database server for parameter management' (checked). Below these are text input fields for 'Hostname' (containing 'aproltraining'), 'Cluster name' (containing 'aproltraining'), 'Domain name' (containing 'br-automation.co.at'), 'Operator bus Network address' (containing 'aproltraining'), and 'Process bus Network address' (containing 'aproltraining'). There is also a dropdown menu for 'Physical view for data export' and a button labeled 'KDE settings' at the bottom right.

Рис. 44. Конфигурация управляющего компьютера

Определение **типа компьютера** (станция оператора, сервер и т.д.)

- Назначение созданного ранее **физического представления**
- Ввод **целевой системы**
- Настройка **порта lousys**
- Настройка **сети**
- **Параметры KDE** (выбор меню для СРЕДЫ ВЫПОЛНЕНИЯ или рабочей среды оператора)
- Создание одной или нескольких задач компьютера

- Настройка модулей системы управления процессами (СС модули)

Примечание. Модули СС

Это конфигурация для приложений и программ, которые будут запущены в исполняющей системе (фоновые программы и программы пользовательского интерфейса)

- Создание *APROL* соединений (*APROL* подключение) и, при необходимости, подключений проекта.

Примечание. Соединения

Оборудование других производителей обычно подсоединяется к компонентам системы управления через GatewayServer. GatewayServer – это независимый компьютер.

- Добавление внешнего оборудования для аппаратного входа в систему.
- Использование в качестве сервера протокола, сервера документации и управляющего компьютера по умолчанию
-

7.2.1. Назначение физического представления

Назначение физического представления однозначно определяет

управляющий компьютер, выполняющий управление в проекте.

Кнопка «**Physical View**» (физическое представление) используется для открытия дерева физического представления. По дереву можно перемещаться и выбирать физическое представление. Выбор осуществляется двойным щелчком или нажатием «**OK**».

7.2.2. Определение типа управляющего компьютера

Помимо чистых систем операторов, которые интегрируются в проект в качестве станций оператора, в списке выбора в различных классах производительности представлены типы **R**, **E**, **O**, **Red** и их комбинации, а также серверы-шлюзы.

Сокращения R, E, O и Red обозначают:

R	Исполняющая система
E	Техническая система
O	Оператор
Red	Резервный сервер
Gateway server	подключение к внешней системе через отдельный сервер.

Выбор компьютера из списка «**Computer type**» (тип компьютера) создает его в системе. Область настройки изменяется при изменении типа, что означает потерю всех ранее сделанных изменений.

Примечание:

При изменении типа компьютера параметры в области настройки теряются, но предыдущая версия сохраняется со всеми параметрами в системе управления версиями.

7.2.3. Выбор исполняющей системы

На одном управляющем компьютере можно настроить исполняющие системы и системы операторов для нескольких проектов.

«**Целевая система**» для исполняющей системы или системы оператора проекта должна быть указана в поле ввода с тем же самым именем, чтобы созданные пользовательские программы передавались с сервера разработки в исполняющие системы проекта или системы операторов других управляющих компьютеров.

Основная программа сбора данных и распространения «**losys**» имеет **коммуникационные порты** для возможности разделения проектов в подобном типе конфигурации. По умолчанию коммуникационный порт 0 следует ввести в поле ввода с тем же самым именем. Если один сервер используется для нескольких приложений, то необходимо, чтобы losys порт различался в разных проектах.

7.2.4. Настройка сети

На вкладке «**Network**» (сеть) для каждого управляющего компьютера необходимо ввести уникальные «**Host Name**» (имя главного компьютера) и «**Operator bus Network address**» (сетевой адрес шины оператора), т.к. компьютеры должны взаимодействовать друг с другом при помощи системной / операторской шины.

Поскольку исполняющая система использует **шину обработки данных** для получения информации, которую необходимо архивировать и передавать далее в систему оператора для визуализации, а также учитывая, что системы разработки используют шину для загрузки на контроллер, шина обработки данных должна быть подробно определена.

Если шина обработки данных проектируется как сеть Ethernet, то она будет отделена от системной шины за счет использования отдельной сетевой платы. В данном случае необходимо ввести сетевой адрес сетевой платы шины обработки данных.

Примечание. Системная шина

Если шина обработки данных и системная шина располагаются на одном и том же сегменте сети, то для обоих используется одинаковый адрес.

«**Default control computer**» – это центральная **система сбора данных**. Главный компьютер взаимодействует со всеми существующими станциями операторов, контроллерами и внешними системами. Более того, для сервера разрабатывается одна или несколько **задач управляющего компьютера**.

Сервер разработки – это центральная **система генерации данных и распространения**.

На ней создаются все пользовательские программы. Затем программы распространяются по системной шине и шине обработки данных на остальные компоненты системы управления процессами и контроллеры.

7.2.5. Создание задач управляющего компьютера

Так же, как и контроллер, управляющий компьютер выполняет свои задачи в однозадачном режиме. Также возможен многозадачный режим.

Одна или несколько **задач управляющего компьютера** могут быть заданы только управляющему компьютеру с уже установленной исполняющей системой.

Задача управляющего компьютера создается при помощи кнопки «**Control computer task**» (задача управляющего компьютера). Задаче необходимо задать уникальное имя и по возможности выразительное описание. Допускается перенастройка пользователем значения по умолчанию времени цикла, равного 1000 мс.

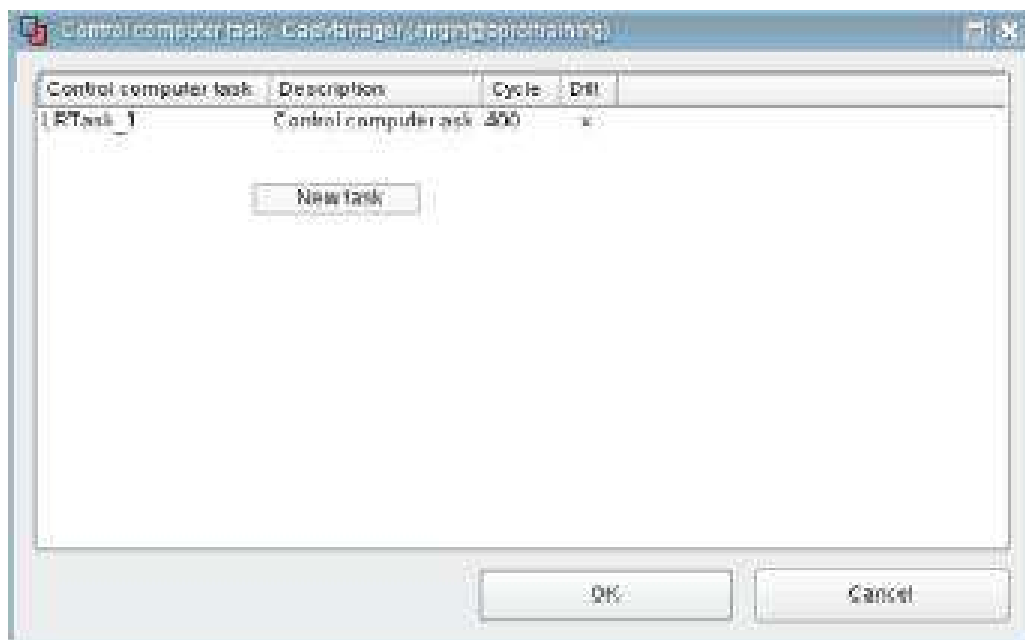


Рис. 45. Создание задачи управляющего компьютера

7.2.6. Настройка модулей системы управления

Благодаря высокой степени гибкости системы *APROL* допускается разработка большого количества конфигураций. Некоторые приложения должны запускаться автоматически при запуске системы, другие должны быть готовы к запуску. Некоторые системы – даже при технической «готовности» – не должны запускаться никогда. Помимо параметра принятия решений о возможности запуска приложения на выбранном компьютере («**Startup**»), существует возможность настройки автоматического запуска приложения при перезапуске системы или входе в рабочую среду операторов («**Auto Startup**»).

В некоторых приложениях существуют **параметры**, детальное описание которых выходит за рамки настоящего семинара. Подробные сведения о параметрах для модулей системы управления см. в «**Руководстве по настройке**» или в меню «**Справка**» соответствующего приложения.

7.2.7. Выбор модулей системы управления

При щелчке на вкладке «**CC Modules**» (модули управляющего компьютера) отображается стандартная сводка приложений для выбранного типа компьютера.

Не допускается удаление модулей, отмеченных x в столбце «Dflt», т.к. их наличие в системе абсолютно необходимо.

Для любого выбранного приложения можно установить **параметры** с помощью контекстного меню (или двойного щелчка кнопкой мыши).

Установка **further instances** (последующих экземпляров) необходима, например, для подключения шины обработки данных, т.к. требуется запускать драйвер (InaDriver) для каждого сетевого подключения PLC.

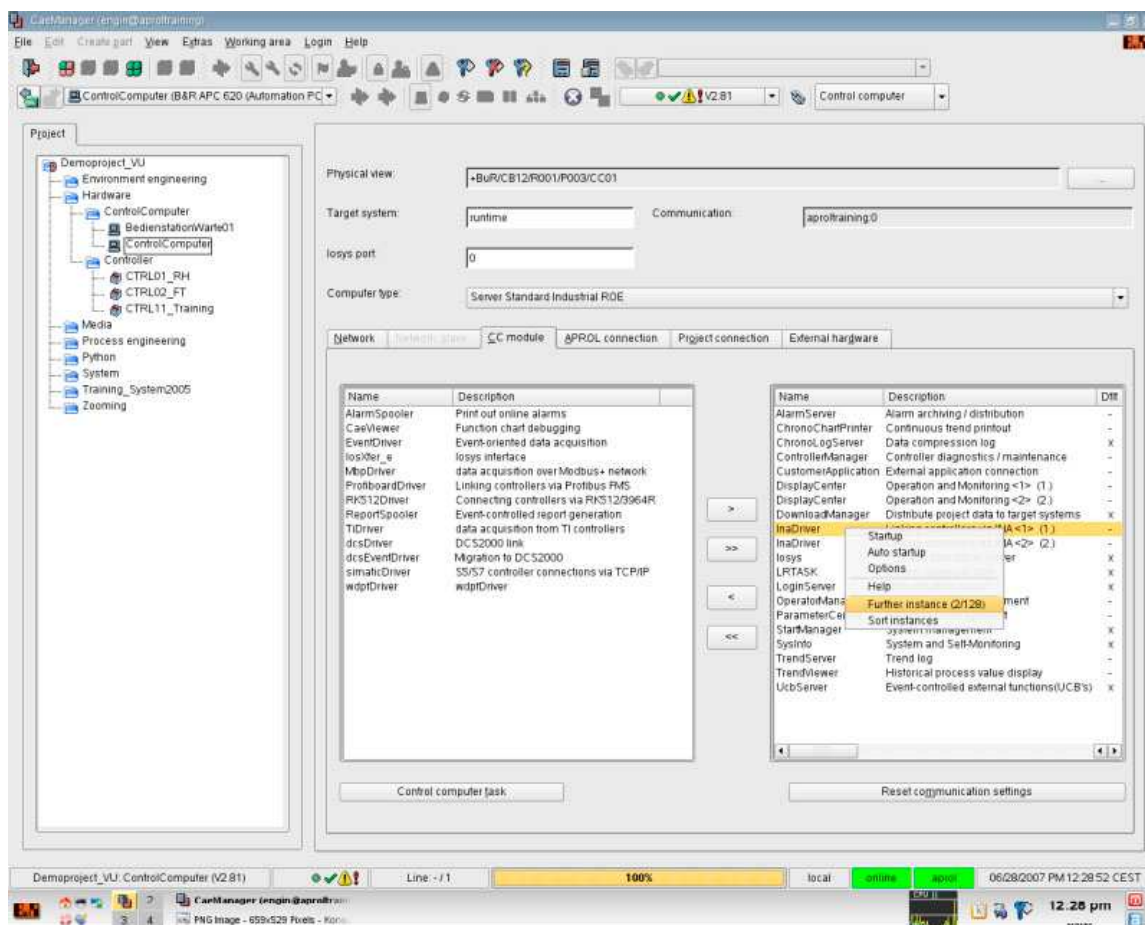


Рис. 46. Модули системы управления

Примечание. Конфигурации станций операторов

Если, например, несколько терминалов операторов должны быть настроены одинаково или почти одинаково, рекомендуется полностью

настроить компьютер, чтобы можно было произвести копирование в CaeManager в окне проекта.

7.2.8. Подключения *APROL*

Подключения к внешним устройствам и другим системам могут быть выполнены не только с помощью контроллеров, но и непосредственно на сервере среды выполнения или через сервер-шлюз. Все драйверы, которые должны быть запущены, сначала необходимо настроить в области модулей СС, также должны быть назначены «StartUp». Параметры для этих подключений, такие как списки переменных и т.д., могут быть установлены в области «**APROL-подключение**».

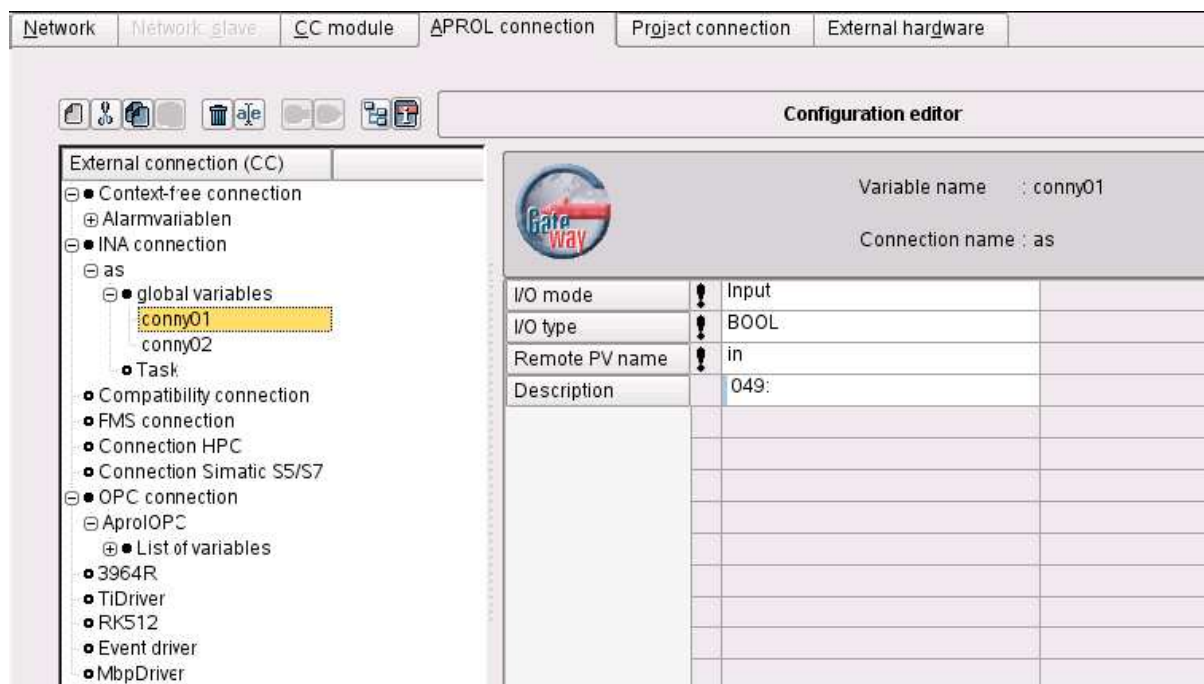


Рис. 47. Подключения APROL

Самые распространенные типы подключений:

- INA подключение к контроллерам B&R (программируемым в AS)
- Подключение Simatic S5/S7
- Подключение OPC
- ...

Сведения о них будут представлены в последующих курсах. Воспользовавшись соответствующей документацией по курсу обучения и интерактивной справкой *APROL*, вы сможете реализовать указанные типы подключений.

7.2.9. Аппаратный вход в систему

С помощью параметров вкладки «**External hardware**» (внешнее оборудование) можно определить, какие аппаратные средства можно использовать для входа пользователя в ранее настроенную систему оператора или исполняющую систему управляющего компьютера.

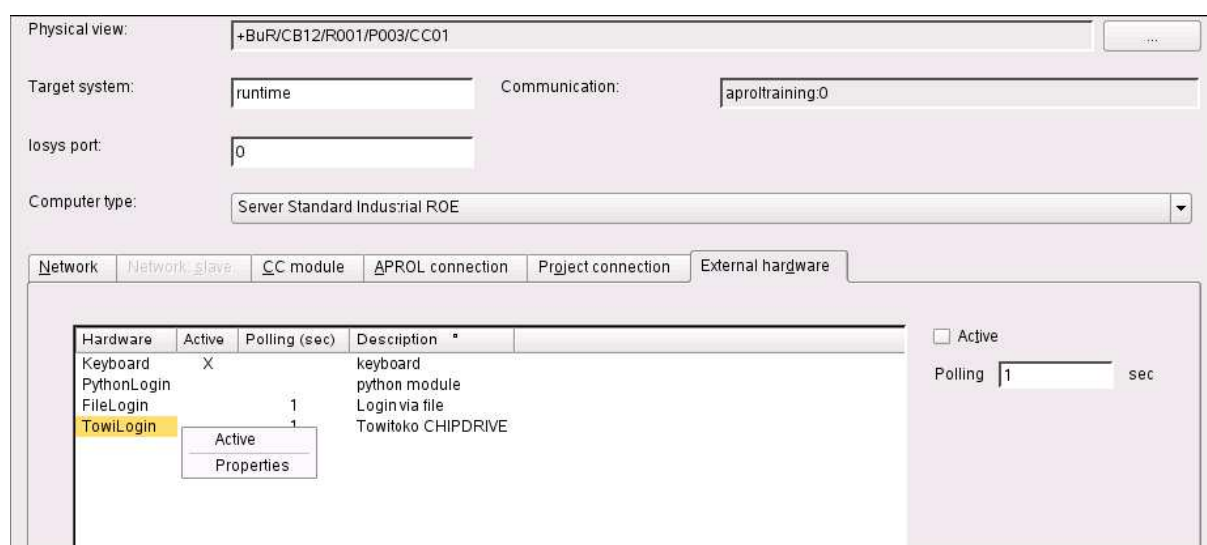


Рис. 48. Настройка аппаратного входа в систему

Для входа пользователя в систему предлагается использовать **Keyboard**, **Python module**, **FileLogin** и **Towitoko CHIPDRIVE**. Для параметров **FileLogin** и **TowiLogin** необходимо задать циклическое время опроса (в секундах) для сервера входа.

Выбранные устройства могут быть активированы, и в контекстном меню (щелчок правой кнопки мыши) для них могут быть настроены дополнительные свойства (если присутствуют). Активация может также быть выполнена при установке флажка «**Active**».

7.2.10. Создание страниц комментариев

Для созданных объектов пользователи могут вставлять страницы

комментариев. При этом предоставляется подробное описание объектов и документация по разработке. Эти

страницы с комментариями располагаются в том же каталоге, что и описываемые в них объекты, что позволяет напрямую их согласовывать.

Страницы комментариев для управляющего компьютера можно создавать и изменять так же, как было описано ранее для контроллеров.

На представлении проводника размещается значок, определяющий наличие страницы с комментариями для объекта.

Изменения на странице с комментариями подпадают под действие управления версиями точно так же, как и изменения соответствующего объекта.

7.2.11. Сохранение изменений

После завершения настройки изменения сохраняются при помощи пункта меню «**File / Save**» (файл / сохранить). При этом, а также при других изменениях генерируется и сохраняется новая версия. Помимо этого,

пользователь может добавить **комментарий** к этой информации. Кнопка «**New Release**» (новый выпуск) позволяет увеличить на единицу номер выпуска.

Управляющий компьютер сохраняется так же, как и контроллер.

7.2.12. Активация и компиляция

Отдельные версии объекта можно сохранить в базе данных разработки *APROL*.

Чтобы указать системе, какие из сохраненных версий следует использовать для генерации и последующего использования проекта, определенная версия **активируется**. Активация версии должна производиться в представлении версий объекта в области разработки CaeManager. Перейти в представление версий можно, выбрав пункт меню «**Versions**» (версии) в контекстном меню объекта (открывающемся при щелчке правой кнопки мыши). Если требуется активировать последнюю сохраненную версию, выберите пункт контекстного меню объекта «**Last version**» (последняя версия).

«**Compile**» (компиляция) проверяет возможность **генерации** объекта. Любые сделанные настройки объекта проверяются на достоверность и правильность синтаксиса. Процесс компиляции выполняется в представлении версии (контекстное меню объекта) точно так же, как активация.

Управляющий компьютер активируется и компилируется так же, как и контроллер.

Упражнение 4

Введите управляющий компьютер в качестве фиктивной станции оператора. Еще раз обратите внимание на удобство чтения проекта. Настройте модули стандартной системы управления.

Место для подробного изложения:

8. КОНФИГУРИРОВАНИЕ КОНТРОЛЛЕРА С ПРОСТОЙ ЛОГИКОЙ

8.1. Общая информация о структурных схемах

Система *APROL* запрограммирована на объектно-ориентированном, графическом языке функциональных блоков в соответствии с IEC 1131-3. Для создания структурных схем доступны следующие типы подпрограмм:

Функции Логический компонент, созданный на языке ANSI C, который обрабатывается в цикле и чье возвращаемое значение называется точно так же, как и сама функция. Вследствие этого функция может возвращать только одно значение (выходное значение).

Функциональные блоки Логический компонент, созданный на языке ANSI C, который может возвращать несколько значений (выходных значений).

Блоки картинок Интерфейс между логическим компонентом и диаграммой процесса. Блоки картинок снабжаются макросами картинок.

Функциональные блоки систем управления С помощью функциональных блоков, предоставляемых компанией B&R, настраиваются функции системы управления, такие как сигнализация и система передачи сообщений.

Макросы (гипер-макросы) Структурные схемы, которые могут размещаться как функциональные блоки. Контакты, выходящие из схем ввода / вывода на функциональном блоке, могут подключаться к окружающей логике.

Блоки UCS Блок пользовательских настроек: в основном запрограммирован на языке Python или скриптах оболочки.

Типы подпрограмм, размещенных на структурной схеме, могут отличаться по цвету.

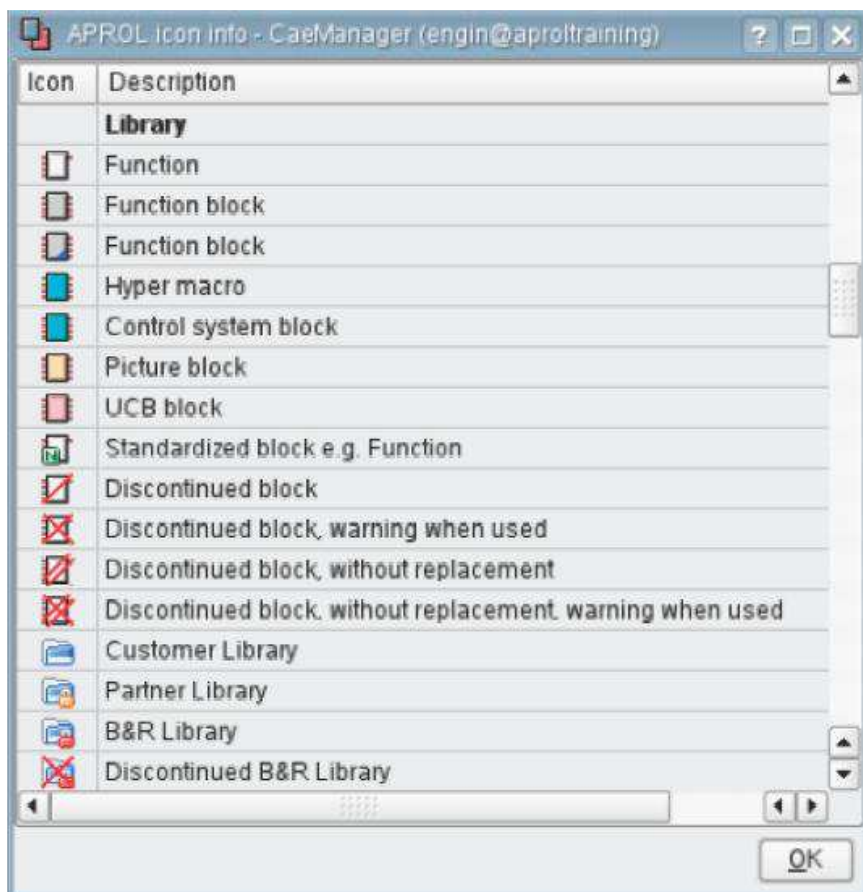


Рис. 49. Типы блоков

Эти функциональные блоки могут размещаться и подсоединяться в структурной схеме с помощью перетаскивания.

8.2. Создание структурной схемы

Структурная схема может быть создана в выбранном каталоге при помощи пункта меню «**Create part / Function chart**» (создать элемент / структурная схема).

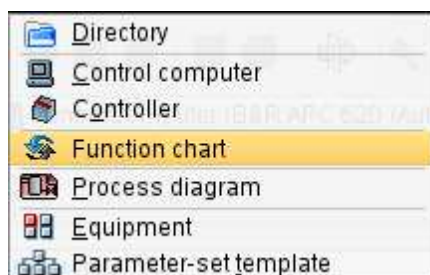


Рис. 50.
Создание структурной схемы

Уникальное имя, выразительное описание, имя экземпляра и физическое представление (задача B&R или задача управляющего компьютера) для новой структурной схемы должны быть введены или выбраны в соответствующих полях.

Имя экземпляра - это краткое обозначение, которое используется системой для формирования имен переменных процесса, настроенных в структурной схеме.

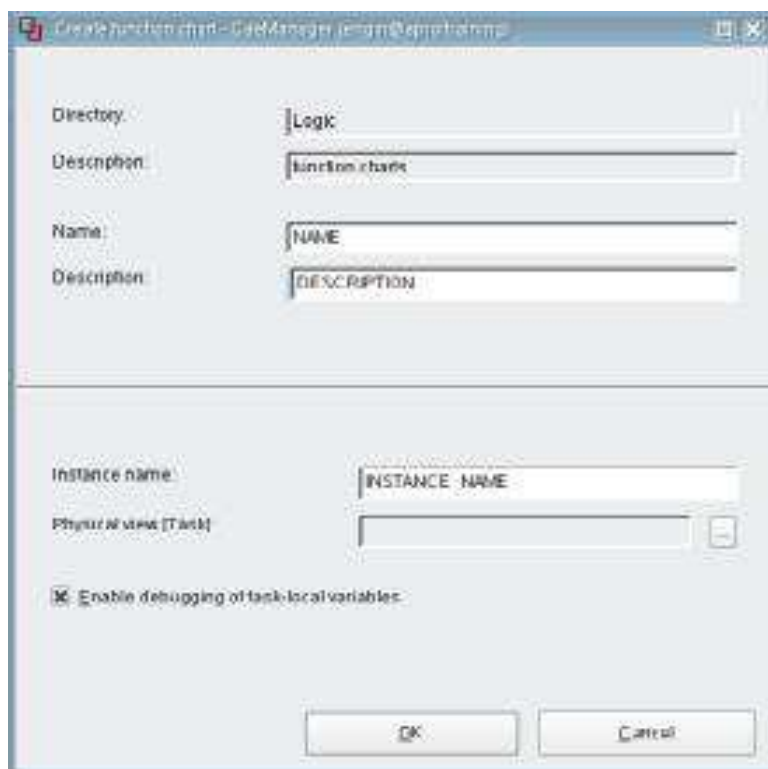
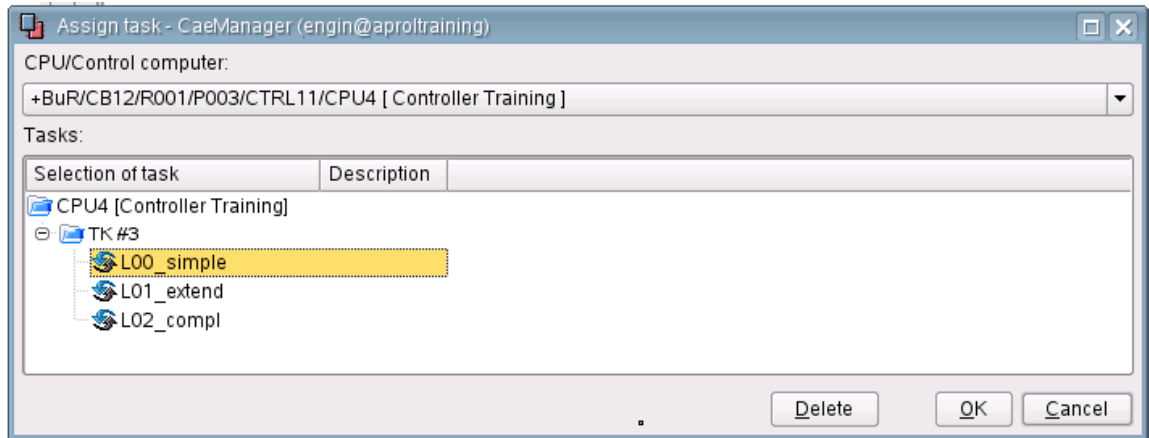


Рис. 51. Новая структурная схема

Рис. 52. Назначение физического представления (задачи)



Внимание! Система обозначений

Имя экземпляра должно начинаться с буквы и быть уникальным в системе (например, DF1, A2). Длина имени не должны превышать 15 символов. Т.к. имена важны при генерации задачи, они должны соответствовать соглашениям языка C. Использование цифр без букв, немецких символов (ä, ö и т.д.) и специальных символов не допускается! Проверка, включенная в *APROL*, также не допускает этого.

8.3. Свойства структурной схемы

Свойства только что созданной структурной схемы (например, описание, экземпляр, назначение физического представления, тип и размер) можно изменить позднее.

Кнопки «**Properties**» (свойства) пунктов контекстного меню можно использовать для открытия окна «Function chart properties» (свойства

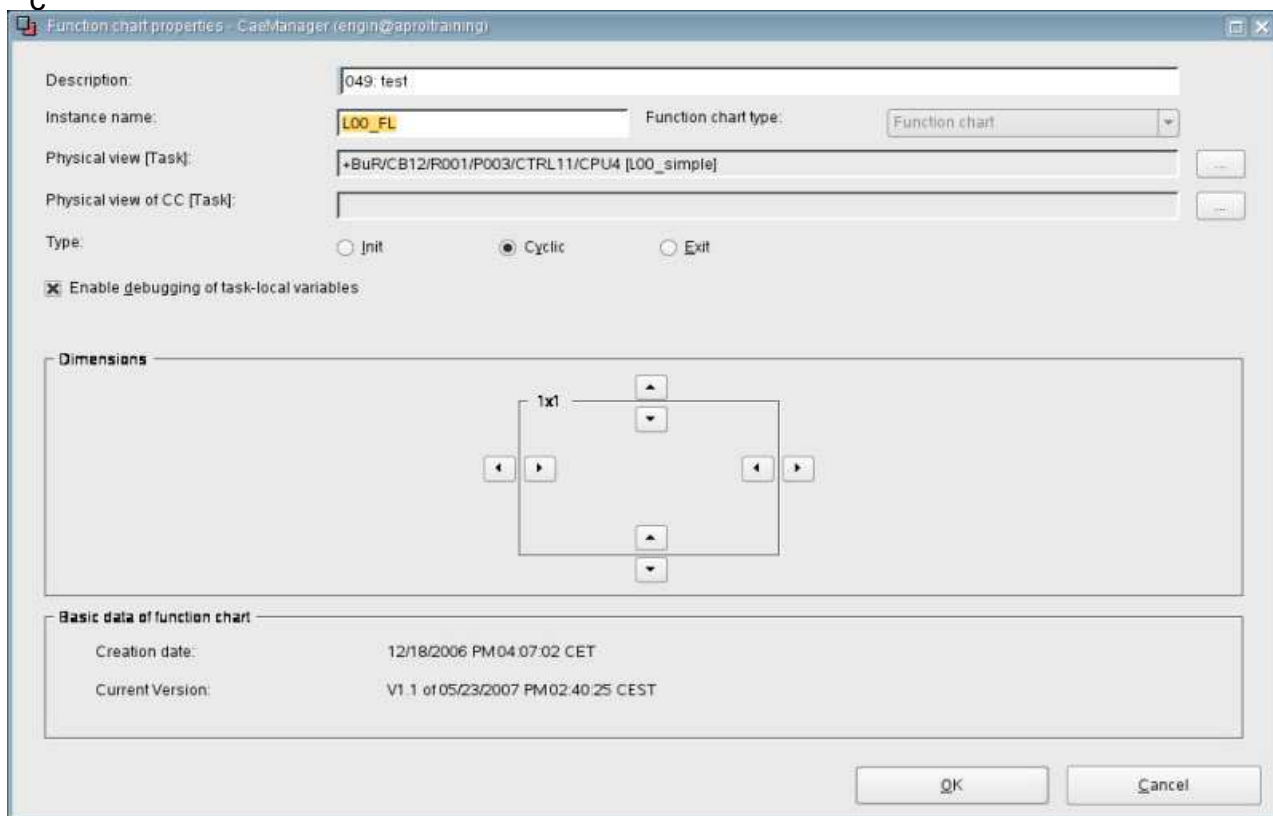


Рис. 53. Свойства структурной схемы

Примечание. Свойства

Свойства можно вызвать в любое время, нажав среднюю кнопку мыши.

Свойство «**Type**» (тип) определяет способ и время выполнения структурной схемы.

- **Init** - Структурная схема выполняется однократно при запуске контроллера.

- **Cyclic** - Структурная схема выполняется циклически.

- **Exit** - Структурная схема выполняется однократно при остановке контроллера или удалении задачи.

Вы можете использовать кнопки в области «**Dimensions**» (размеры) для изменения размеров структурной схемы. Изменять размеры допускается во всех направлениях, даже после начала разработки. Уменьшение размера схемы может быть выполнено только на страницах без функциональных блоков.

Помимо этого, возможно изменить **физическое представление**, выбирая соответствующие кнопки.

Примечание. Изменение физического представления

Изменение физического представления структурной схемы не подпадает под назначение версий. В результате данное изменение применяется ко всем версиям объектов.

Вторая возможность выбора используется для назначения задачи управляющего компьютера в случае использования нескольких задач в проекте (многозадачность управляющего компьютера).

Примечание. Расширенная структурная схема

Максимальный размер структурной схемы – 10 x 10 страниц. Схема может быть расширена в любом направлении, включая вверх и влево, что очень важно при редактировании. Существующие линии связи продлеваются автотрассировщиком.

8.4. Навигация в редакторе структурных схем

Большая часть первой страницы структурной схемы отображается на стандартном экране редактора структурных схем. Чтобы просмотреть все части первой страницы, схему можно перемещать слева направо с помощью нижней полосы прокрутки. Разрешение экрана определяет возможную степень отображения схемы. Рекомендуется использовать максимальное разрешение, чтобы на экране отобразилась вся схема (если она занимает одну страницу).

Для просмотра страницы целиком включите полноэкранный режим. Выберите пункт меню «**View / Fullscreen**» (вид / полный экран) или нажмите соответствующую кнопку на панели инструментов. Таким же способом можно и отключить данный режим.

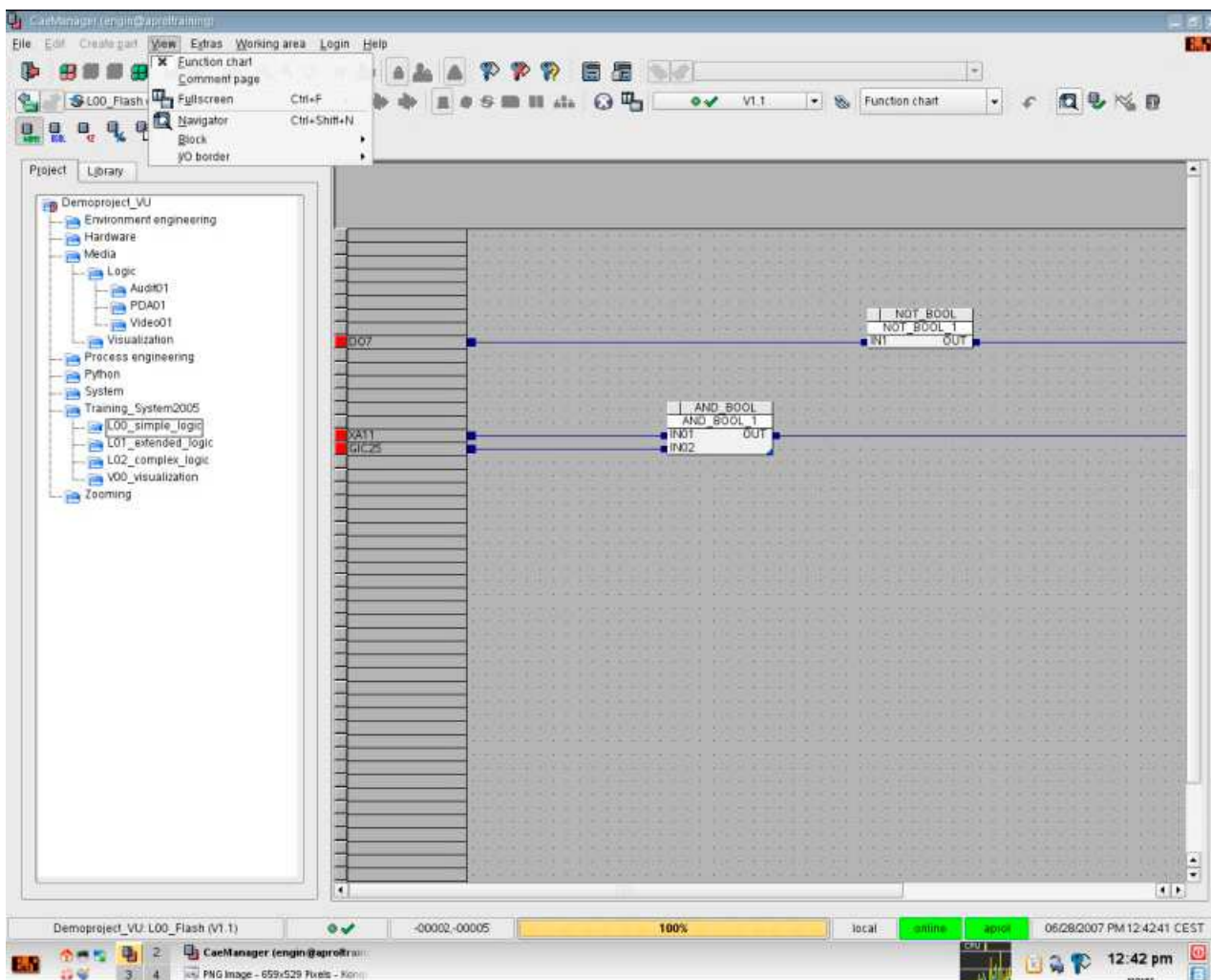


Рис. 54.
Активирование навигации в структурной схеме

Если схема занимает несколько страниц, для навигации можно использовать следующие возможности:

- Выберите пункт меню «**View / Navigator**» (вид / навигатор) или нажмите соответствующую кнопку на панели инструментов.



Рис. 55. Активирование навигатора

Выберите и перемещайте выделенную рамку в активированном навигаторе. Уже настроенные блоки отображаются в данном окне в символической форме разными цветами.

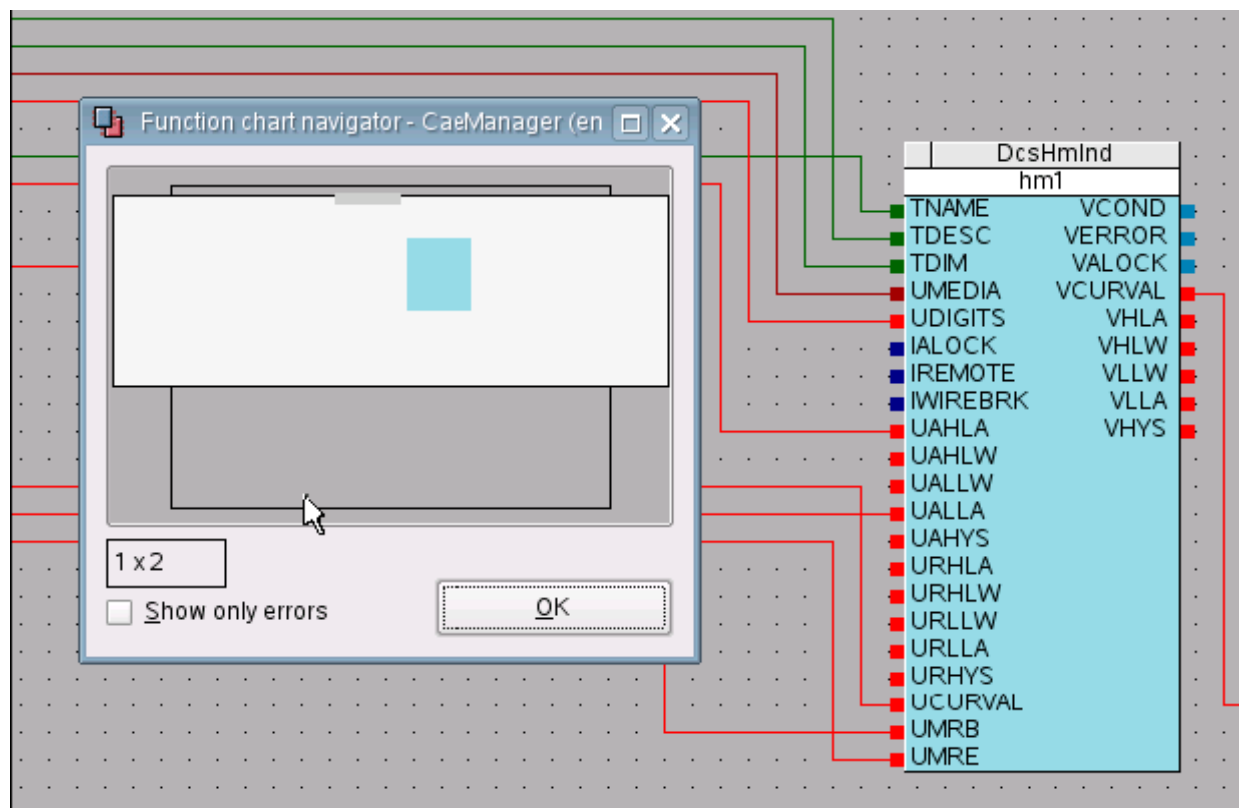


Рис. 56. Навигация

- Пункт меню «**View / Navigator**» (вид/ навигатор) либо не выбран, либо можно отменить выбор из панели инструментов:

Нажмите клавишу CTRL и «перетащите» структурную схему в требуемую область.

Структурная схема состоит из матрицы страниц формата DIN A4 максимального размера 10 x 10 страниц. Линия стыка со следующей страницей выделяется пунктирной линией. Первый столбец матрицы страниц содержит рамку ввода. В последнем столбце содержится рамка вывода.

8.5. Функциональные блоки

8.5.1. Вставка блоков

Чтобы поместить логические функциональные блоки, необходим доступ к библиотекам функциональных блоков, связанных с проектом. Доступ к библиотекам осуществляется из вкладки «**Library**» (библиотека) в области навигации SaeManager. Требуемые блоки выбираются из дерева библиотеки и размещаются на структурной схеме при помощи «перетаскивания». Также можно открыть **справочную информацию** для функционального блока. Пункт меню «**References**» (ссылки) используется для вывода всех элементов проекта, в которых содержится выбранный функциональный блок, а также для отображения места расположения элементов (источник). С помощью данной функции пользователь также может непосредственно перейти по отдельным ссылкам.

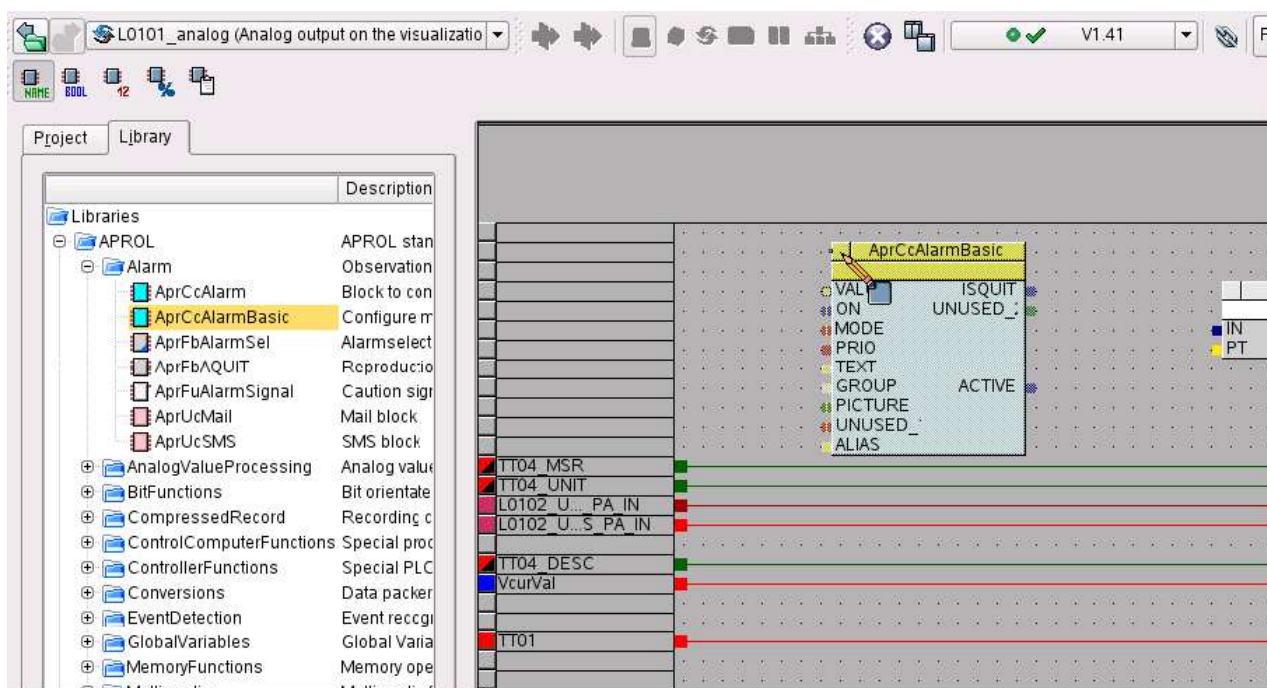


Рис. 57. Вставка функциональных блоков

Примечание. Удобство чтения FC

Разместите функциональный блок таким образом, чтобы входные и выходные линии по возможности не пересекались и были четко структурированы. Это упрощает чтение структурной схемы.

Функциональным блокам необходимо дать уникальные имена (экземпляры), т.к. одни и те же функциональные блоки можно использовать в структурной схеме многократно. При размещении на структурной схеме функциональному блоку автоматически дается имя. После этого имя типа функционального блока подчеркивается и нумеруется. Каждый раз при размещении функционального блока одного и того же типа номер увеличивается на 1. В большинстве случаев понятно, когда экземпляр функционального блока изменялся вручную по назначенному логическому имени, характерному для системы (TAG, MSR).

Примечание. Именованное экземпляра

Имя функционального блока данного типа: Controller_A. Первый экземпляр данного типа функционального блока в схеме: «Controller_A_1». Второй экземпляр данного типа функционального блока в схеме: «Controller_A_2» и т.д. Автоматически назначаемое имя можно изменить в окне «**Properties**» (свойства) функционального блока или щелкнув средней кнопкой мыши по функциональному блоку. Выбирайте наиболее короткое имя, т.к. имена переменных строятся на основании этого имени.

Примечание. Именованное

Функциональным блокам, часто используемым в логике, для упрощения отладки необходимо давать уникальные имена (имена тегов), связанные с функцией.

При вставке функционального блока его копия в действительности не вставляется в структурную схему, создается только ссылка в базе данных и предоставляются индивидуальные параметры (имя экземпляра, константы и т.д.). В объектно-ориентированных языках программирования это может означать, что: **создается и настраивается экземпляр указанного класса (функциональный блок).**

На данном этапе важно отличие копии от экземпляра, т.к. это влияет на работу измененных функциональных блоков. **При изменении класса изменяется порождаемый экземпляр.**

Таким образом, изменение функционального блока воздействует на все вхождения функционального блока в проект. В случае неизменности функционального блока вместо создания ссылки блок необходимо скопировать (см. документацию *APROL* «Руководство разработчика»). Затем изменения могут вноситься в копию (это означает создание нового функционального блока с новым именем). Сделанные изменения не

применяются до тех пор, пока новый функциональный блок не будет использован в проекте.

После размещения функционального блока в структурной схеме можно открыть его **контекстное меню** (поместив курсор на функциональный блок и щелкнув правой кнопкой мыши).

Затем можно вызывать различные функции (активные записи), зависящие от типа функционального блока.

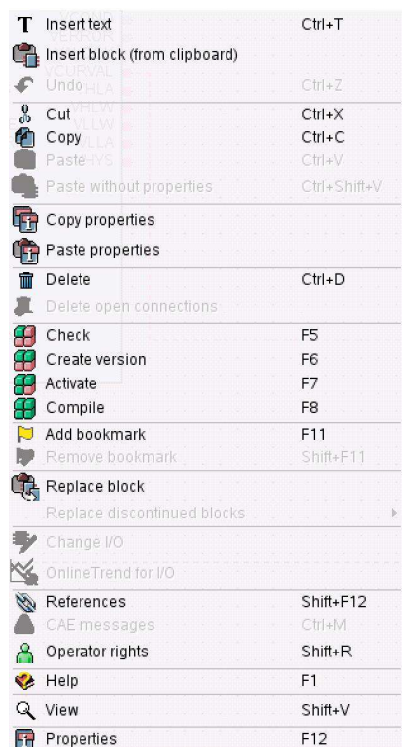


Рис. 58. Меню блока в структурной схеме

8.5.2. Перемещение блоков

В дальнейшем функциональный блок можно переместить. Для этого щелкните и перетащите функциональный блок в требуемое положение.

8.5.3. Представление функционального блока

При помощи гипер-макросов и блоков картинок можно переключаться между представлением функционального блока и структурной схемы за ним или графическим элементом.

Символ стрелки на панели инструментов структурной схемы бывает видим, если вы переключились на представление структурной схемы для функционального блока. Нажатие этой кнопки закрывает представление структурной схемы функционального блока.

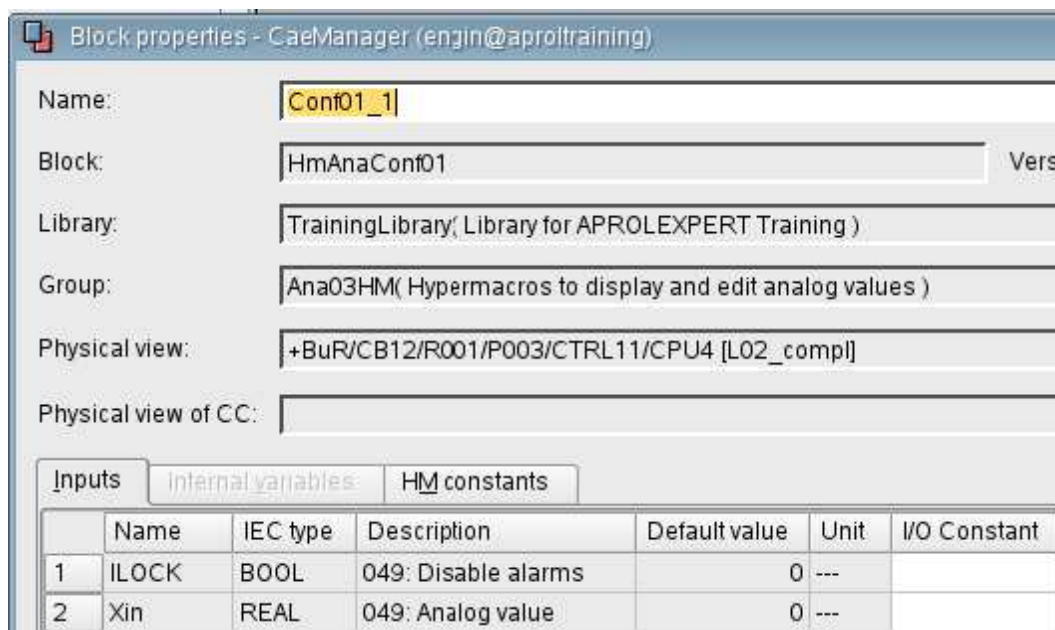
Примечание. Представление гипер-макросов

Как мы увидим позднее, при интерактивной отладке также допускается переходить на внутреннее представление структурной схемы гипер-макросов.

Для всех остальных типов функциональных блоков невозможно выбрать пункт меню «**View**» (вид) (или двойной щелчок).

8.5.4. Свойства функционального блока

Чтобы изменить свойства функционального блока, выберите его, нажмите «**Properties**» (свойства) в контекстном меню (или щелкните средней кнопкой мыши). В данном окне предоставляется информация о функциональном блоке в соответствии со структурной схемой.



Block properties - CaeManager (engin@aproiltraining)

Name: Conf01_1

Block: HmAnaConf01 Vers

Library: TrainingLibrary(Library for APROLEXPRT Training)

Group: Ana03HM(Hypermacros to display and edit analog values)

Physical view: +BuR/CB12/R001/P003/CTRL11/CPU4 [L02_compl]

Physical view of CC:

Inputs Internal variables HM constants

	Name	IEC type	Description	Default value	Unit	I/O Constant
1	ILOCK	BOOL	049: Disable alarms	0	---	
2	Xin	REAL	049: Analog value	0	---	

Рис. 59. Свойства функционального блока

В поле «**Name**» (имя) можно изменить автоматически установленное имя функционального блока (экземпляра).

Поле «**Physical view**» (физическое представление) используется для определения физического представления, которое отличается от структурной схемы для данного функционального блока. Представление удобно использовать, если определенные логические функции расположены на управляющем компьютере высшего уровня.

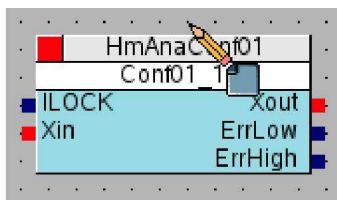


Рис. 60. Измененное физическое представление

Примечание. Изменение физического представления

После выхода из окна в верхней левой части функционального блока появляется красный квадрат, показывающий, что запись физического представления была обработана тем или иным способом.

На вкладке «**Input**» перечисляются все входы функционального блока. Они могут быть подсоединены или соответствующим образом настроены.

Вкладка «**NM Constants**» (константы НМ) не обрабатывается вне функционального блока. На ней назначаются фиксированные значения, которые могут быть изменены только в ходе разработки.

Не допускается соединение различных типов данных. Для этого необходимо использовать конвертер. Для предотвращения ошибок при вводе констант НМ и входов используются валидаторы.

Примечание. Валидатор

В APROL невозможно сделать синтаксическую ошибку при вводе данных. Каждое поле ввода снабжается валидатором для проверки. Красный предупреждающий треугольник отображается в случае ввода некорректной записи, при этом запись невозможно сохранить.

«Help» открывает справку по функциональному блоку. Справка в виде HTML-файла открывается при помощи браузера, настроенного в пункте меню «Extras / Options» (дополнительные возможности / параметры).

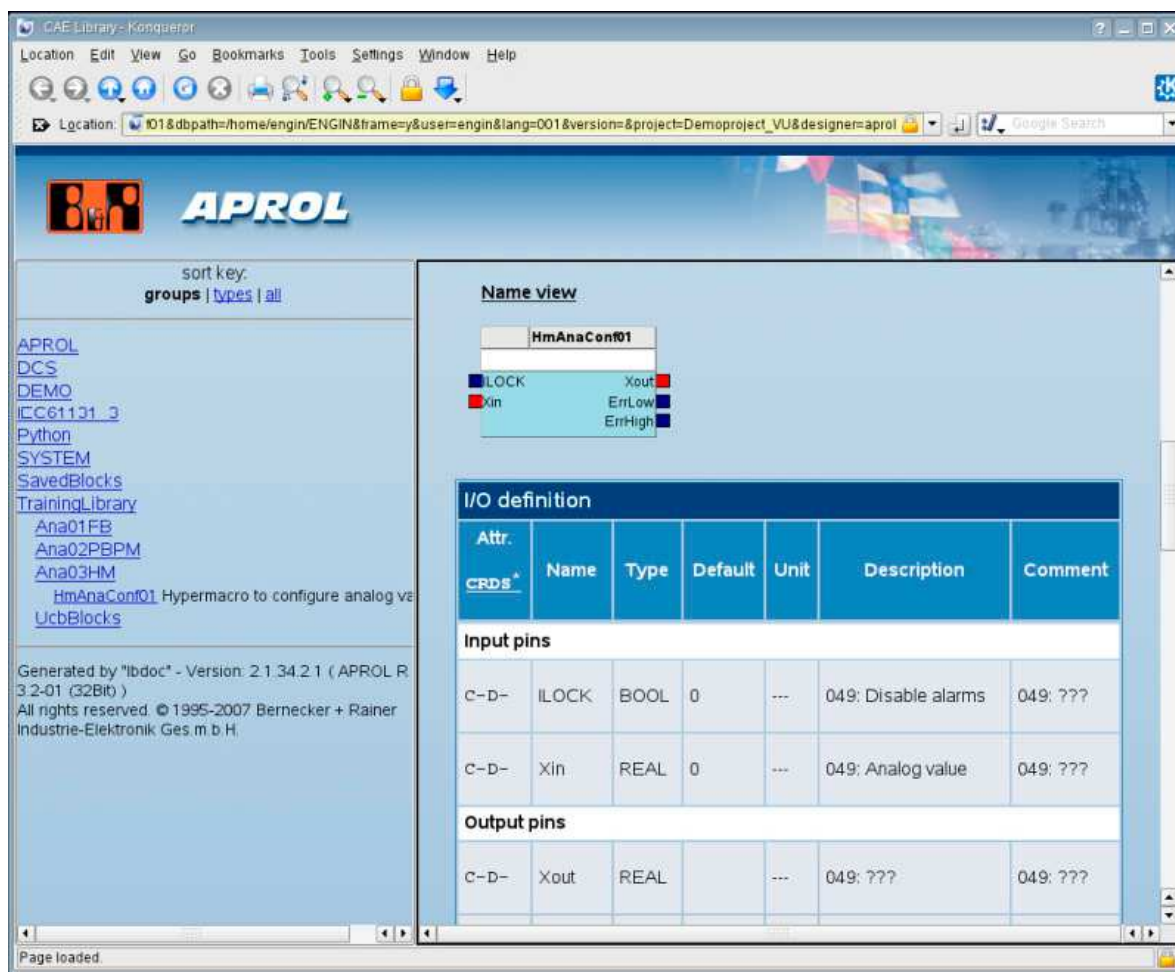


Рис. 61. Справка по блоку

В списке входов функционального блока отображается следующая информация:

Name: Имя входа

IEC type: Тип входных данных.

Description: Описание входа.

Default value: Значение по умолчанию используется в качестве входного значения и дополнительно рассчитывается, если значение не предоставляется или отсутствует заранее определенное значение в записи констант ввода / вывода.

Unit: Единицы измерений / размерности входа

Константы входа / выхода: В качестве входных значений вместо линии питания может задаваться константа, перезаписывающая значение по умолчанию.

В библиотеках, предоставленных компанией B&R, содержатся подробные сведения. Разработчик, проектирующий пользовательские библиотеки, отвечает за составление подобного описания. В большинстве случаев это осуществляется при помощи документации AsBuilt средств разработки библиотек. Каждая сделанная запись и введенный комментарий автоматически помещается в документацию.

8.5.5. Ввод констант

Ввод констант в высшей степени важен для определенных функциональных блоков. Что и почему должно вводиться в виде констант, указано в описании функционального блока (пункт **Help** контекстного меню).

Тип «**Time**» (время) имеет определенный входной формат. Длительность одной секунды может быть введена следующими способами:

1000 интерпретируется как мс

T#1s интерпретируется как секунды (min, h, d и т.д.)

8.5.6. Удаление функциональных блоков

Чтобы удалить функциональный блок, щелкните его правой кнопкой мыши и выберите «**Delete**» (Ctrl+ D).

8.5.7. Текст комментария, заданный непосредственно на структурной схеме

Для удобства чтения структурных схем рекомендуется сохранять комментарии непосредственно на них. Это выполняется при помощи контекстного меню «**Insert text**» (вставить текст). Для ввода текста открывается входной шаблон, затем текст размещается в требуемом положении на структурной схеме.

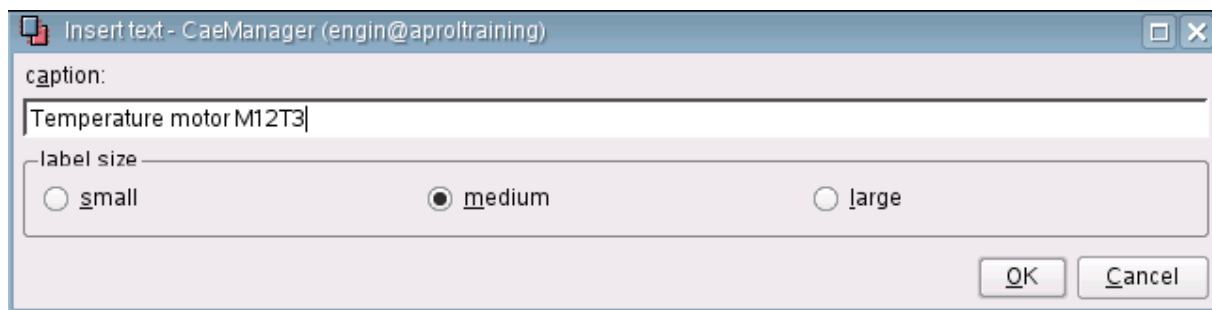


Рис. 62.
Ввод текста структурной схемы

8.6. Соединения

8.6.1. Рисование соединений

Для выбора точек соединения можно использовать простой щелчок левой кнопкой. Автотрассировщик автоматически выберет оптимальный путь между двумя точками. Чтобы ограничить возможные отклонения автотрассировщика, можно щелкнуть левой кнопкой и добавить координаты в схеме. Это удобно, например, для разветвленных длинных линий кабелей.

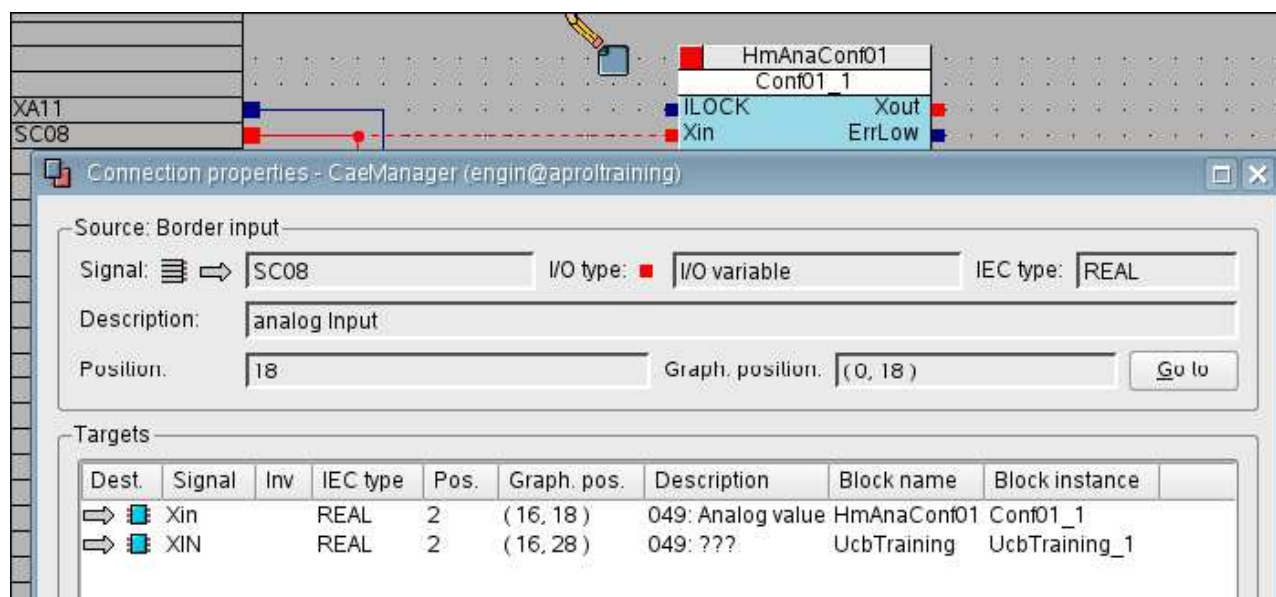
8.6.2. Перемещение соединений

Линия питания перемещается так же, как и функциональный блок. Щелкните линию, удерживайте нажатой кнопку мыши и перемещайте.

8.6.3. Свойства линии питания

Диалоговое окно «**Connection properties**» (свойства соединения) можно открыть щелчком правой кнопки мыши линии питания и выбором функции «**Properties**» (свойства) (или нажатием средней кнопки мыши).

Источник отображается в верхней части диалогового окна, а список целевых объектов для линии питания – в нижней части окна. Если щелкнуть один из целевых объектов или источник в списке, структурная схема автоматически



перейдет в точку их расположения.

Рис. 63. Свойства соединения

8.6.4. Удаление линий связи

Чтобы удалить линию питания, выберите ее, щелкните правой кнопкой мыши и выберите функцию «**Delete**» (удалить).

8.7. Схемы ввода / вывода

Схемы ввода / вывода формируют интерфейс структурной схемы с оборудованием ввода / вывода контроллера, шлюзом ввода / вывода, исполняющей системой, другой структурной схемой, а также параметрами, которые могут быть загружены в систему с помощью ParameterCenter исполняющей системы.

Аппаратные входы, переменные системы связи, разъемы, системные переменные, параметры и системные константы, которые должны быть считаны, вводятся в схеме ввода. Данные, которые должны быть записаны, такие как аппаратные входы / выходы, шлюз ввода / вывода, разъемы и параметры (для загрузки параметров), вводятся в схему вывода. Щелчок правой кнопкой по схеме ввода позволяет выбрать параметры ввода.

При щелчке правой кнопкой по ранее созданной переменной структурной схеме можно удалить вход или открыть «**Properties**» (свойства) перекрестных ссылок этой переменной.

A1

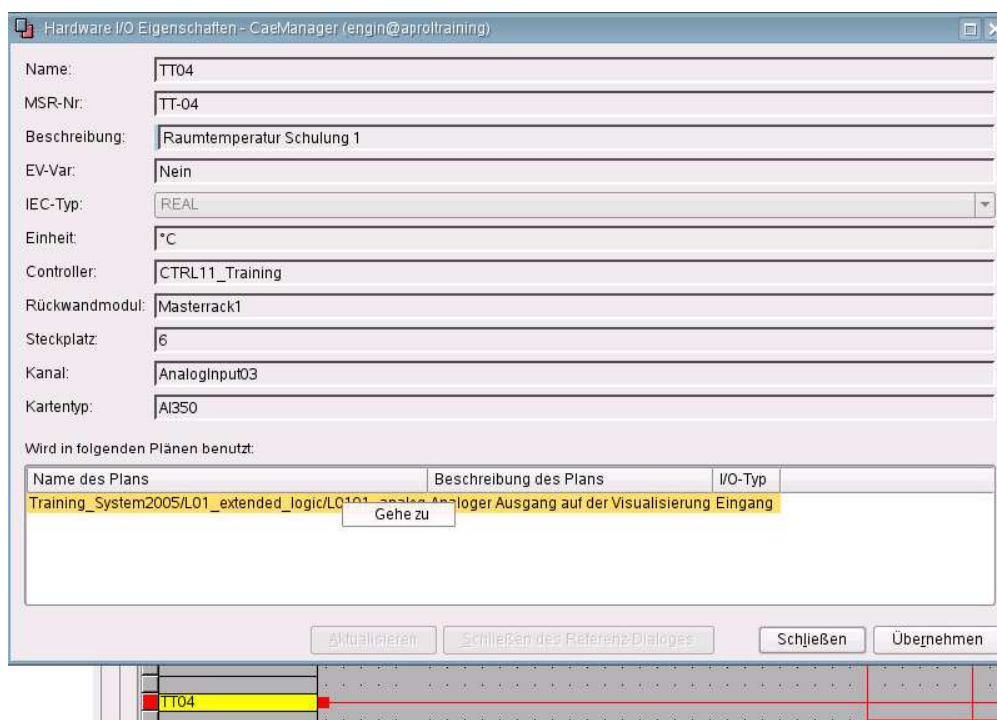


Рис. 64. Свойства входа схемы

Входной контакт – желтого цвета. Щелчок правой кнопки мыши по перекрестной ссылке (или двойной щелчок) сразу же перемещает в соответствующую структурную схему. Запись там также подсвечивается желтым.

Delete удаляет указанный вход, при этом удаляется только его назначение к подключению, а не сама линия связи.

Ввод / вывод **сдвигается** при щелчке и перетаскивании его в требуемую позицию. Линия связи перемещается вместе с вводом / выводом. Удерживайте клавишу Shift для разъединения.

Существует большое количество подсказок для проектирования структурных схем. Для получения сведений о них см. в CaeManager вкладку «CaeManager-Tips» или интерактивную справку *APROL*.

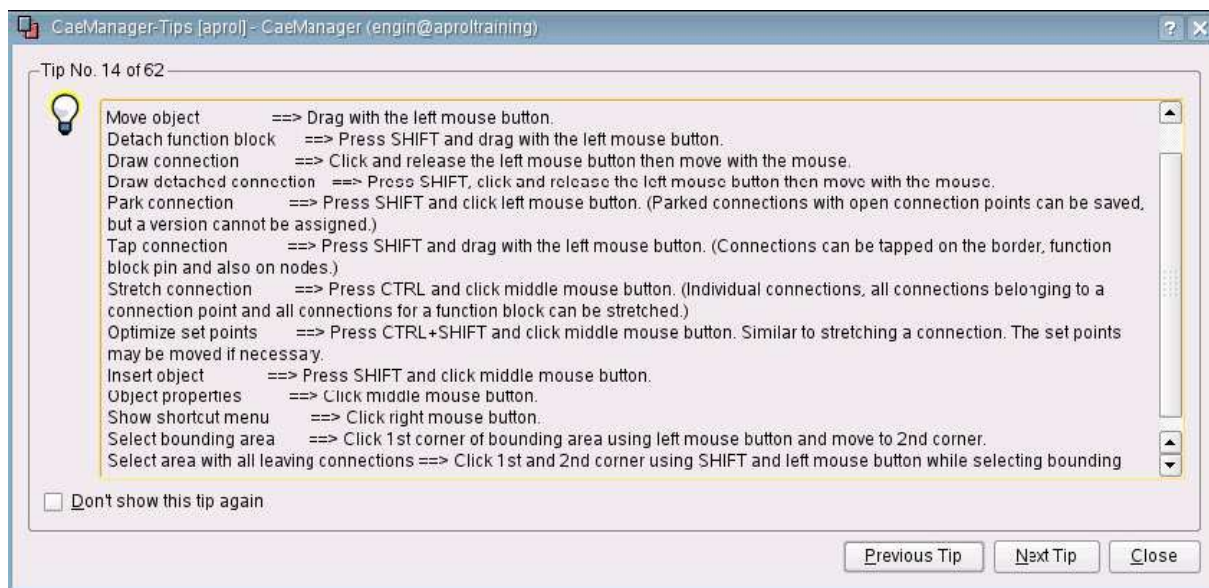


Рис. 65. Подсказки для проектирования структурных схем

8.7.1. Выбор аппаратуры ввода / вывода

В следующем диалоговом окне «Hardware I/Os» перечислены все настроенные аппаратные входы. Используйте данную возможность для ограничения количества данных при помощи задания фильтра в верхней части окна. В полях «Name» (имя), «MSR-Nr» и «Description» (описание) могут использоваться **знаки подстановки "*"**.

Следующие списки выбора ограничены существующими в проекте данными. Например, не имеет смысла искать точки данных на аналоговой плате вывода, если данный тип платы не используется в проекте.

Область «**Use extended I/O information**» (использовать расширенную информацию ввода/ вывода) можно использовать для выбора не только действительных переменных, но также их свойств, описания и состояний.

Состояние входного значения определяется при помощи аналоговых входов, если входной сигнал настроен в диапазоне от +4 мА до +20 мА. В данном случае разомкнутая цепь обнаруживается программным способом. При

возникновении ошибок 1 возвращается как **тип данных WORD**. В противном случае возвращается 0.

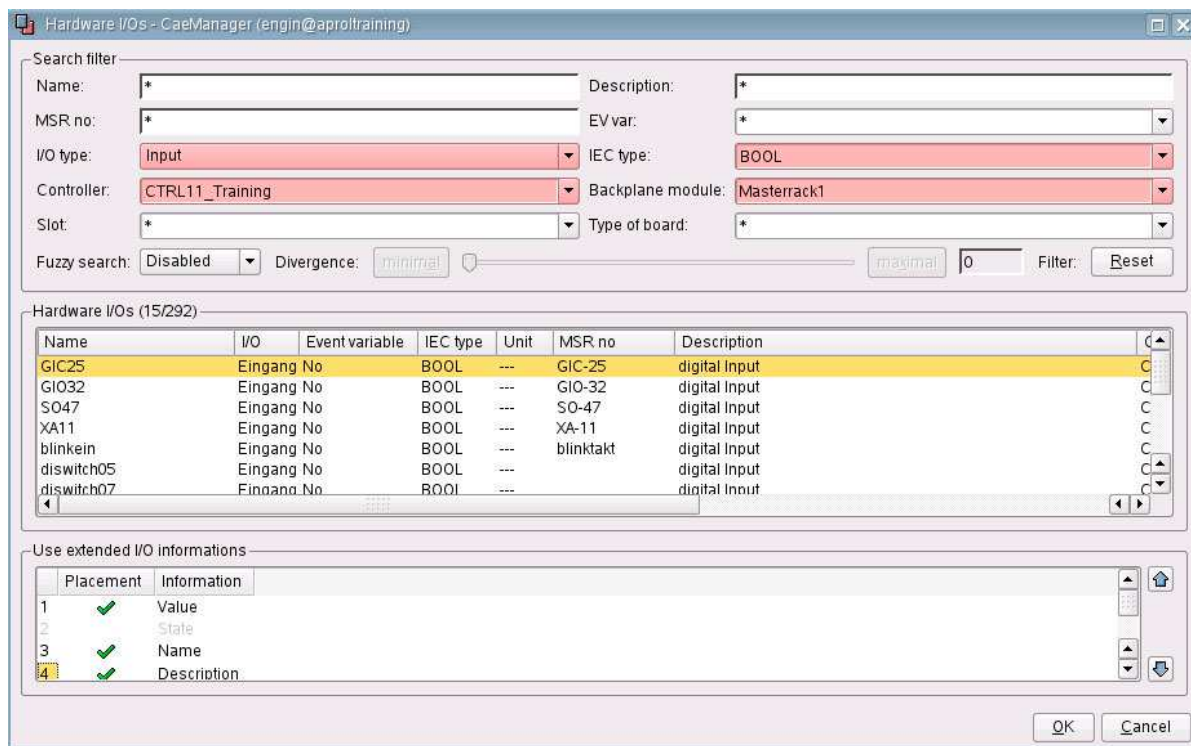


Рис. 66. Вставка аппаратного ввода / вывода

8.7.2. Выбор шлюза ввода / вывода

Соединения в масштабах всей системы устанавливаются при помощи переменных шлюза. В проекте доступны все переменные, определенные в подключении шлюза на управляющем компьютере или контроллере.

Выбор производится так же, как и в случае аппаратного ввода / вывода.

8.7.3. Выбор разъема

Разъем может использоваться для установки соединения при помощи схемы. Разъем создается как локальная переменная задачи и затем преобразуется в глобальную переменную, если создается соединение, выходящее за рамки задачи.

Выбор производится так же, как и в случае аппаратного ввода / вывода.

Если не удастся найти требуемое соединение, допускается создание нового.

Примечание. Локальные и глобальные переменные

Локальная переменная задач использует ячейки памяти контроллера только при выполнении хотя бы одной из ее задач. Глобальная переменная требует выделения ячейки памяти, т.к. ее текущее значение должно также предоставляться и другим задачам.

При помощи кнопки «Create» создается новый разъем на основании ввода данных и добавляется в список выбора. Если разъем уже существует, он становится выбранным. Кнопка «OK» также создает разъем, но при этом закрывается окно и разъем назначается полю на схеме ввода структурной схемы.

Примечание. Разъемы

Убедитесь, что тип связи между входами и выходами разъема – один ко многим (1 к n), т.е. разъем может использоваться только для одного выхода и нескольких входов. И наоборот, разъем следует использовать по крайней мере для одного входа и только для одного выхода.

8.7.4. Выбор системной переменной

Программа, настроенная для исполняющей системы, выполняется на каждом компьютере, настроенном в исполняющей системе (станция оператора 1 к n, исполняющий сервер и главный компьютер шлюза) во время выполнения. APROL автоматически генерирует системные переменные из информации, настроенной в области разработки, эти переменные в дальнейшем предоставляются пользователю для мониторинга.

Некоторые системные переменные предоставляются в типе данных INT и могут расцениваться в соответствии со следующей таблицей.

Информация о значениях

0 – информация о статусе отсутствует

1 – программа выполняется в стандартном режиме

2 – программа выполняется в нештатном режиме

3 – программу не удастся запустить

Системные переменные получают информацию об отслеживаемом сервере или станции оператора. В данном примере используются: Idle Time (время простоя), Disk Space (дисковое пространство), DongleStatus (статус аппаратного ключа) и т.д.

Выбор производится так же, как и в случае аппаратного ввода / вывода.

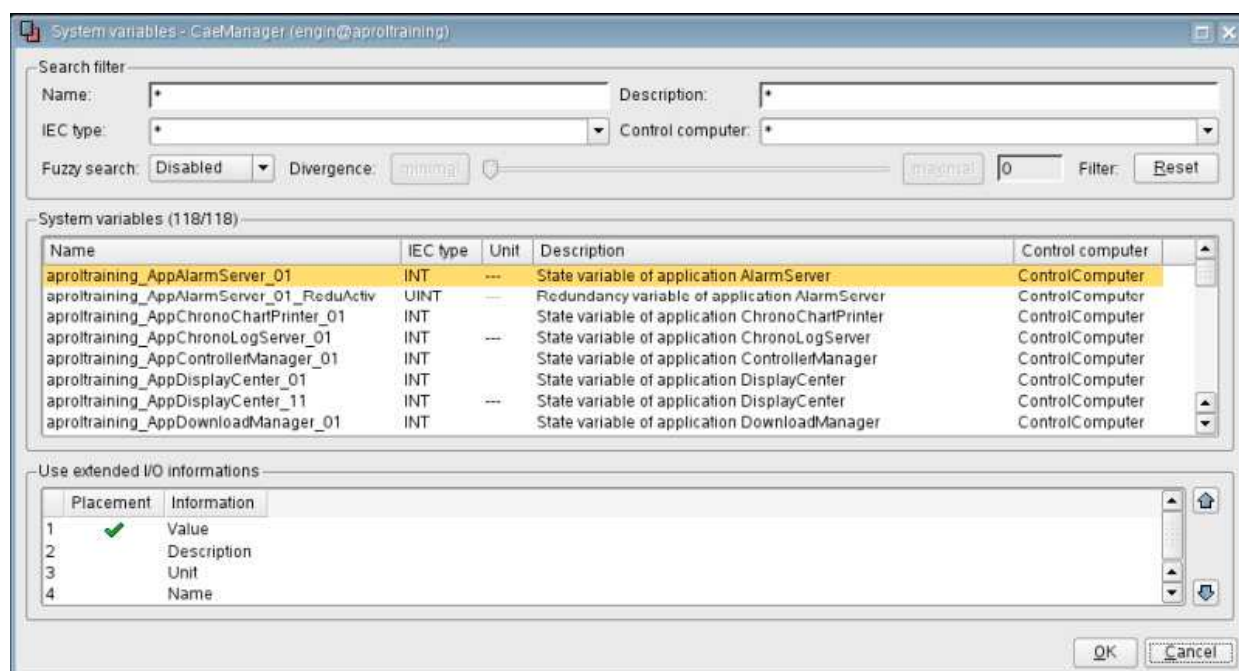


Рис. 67. Выбор системных переменных

Переменные помещаются на схему ввода из выбранного списка при помощи двойного щелчка на переменной и нажатия OK. В дальнейшем она может быть связана.

8.7.5. Выбор параметра

Все переменные, определенные в проекте в качестве параметров, могут быть выбраны. Выбор производится так же, как и в случае аппаратного ввода / вывода.

8.7.6. Выбор системной константы

Все переменные, определенные в проекте в качестве констант, могут быть выбраны.

Выбор производится так же, как и в случае аппаратного ввода / вывода.

8.8. Активация и компиляция структурной схемы

Отдельные версии объекта можно сохранить в базе данных разработки *APROL*.

Чтобы указать системе, какие из сохраненных версий следует использовать для генерации и последующего использования проекта, определенная версия **активируется**. Активация версии должна производиться в представлении версий объекта в области разработки CaeManager. Перейти в представление версий можно, выбрав пункт меню «**Versions**» (версии) в контекстном меню объекта (открывающемся при щелчке правой кнопкой мыши). Если требуется активировать последнюю сохраненную версию, выберите пункт контекстного меню объекта «**Last version**» (последняя версия).

«**Compile**» (компиляция) проверяет возможность **генерации** объекта. Любые сделанные настройки объекта проверяются на достоверность и правильность синтаксиса. Процесс компиляции выполняется в представлении версии (контекстное меню объекта) точно так же, как активация.

Нижe представлена стандартная выходная информация при компиляции структурной схемы.

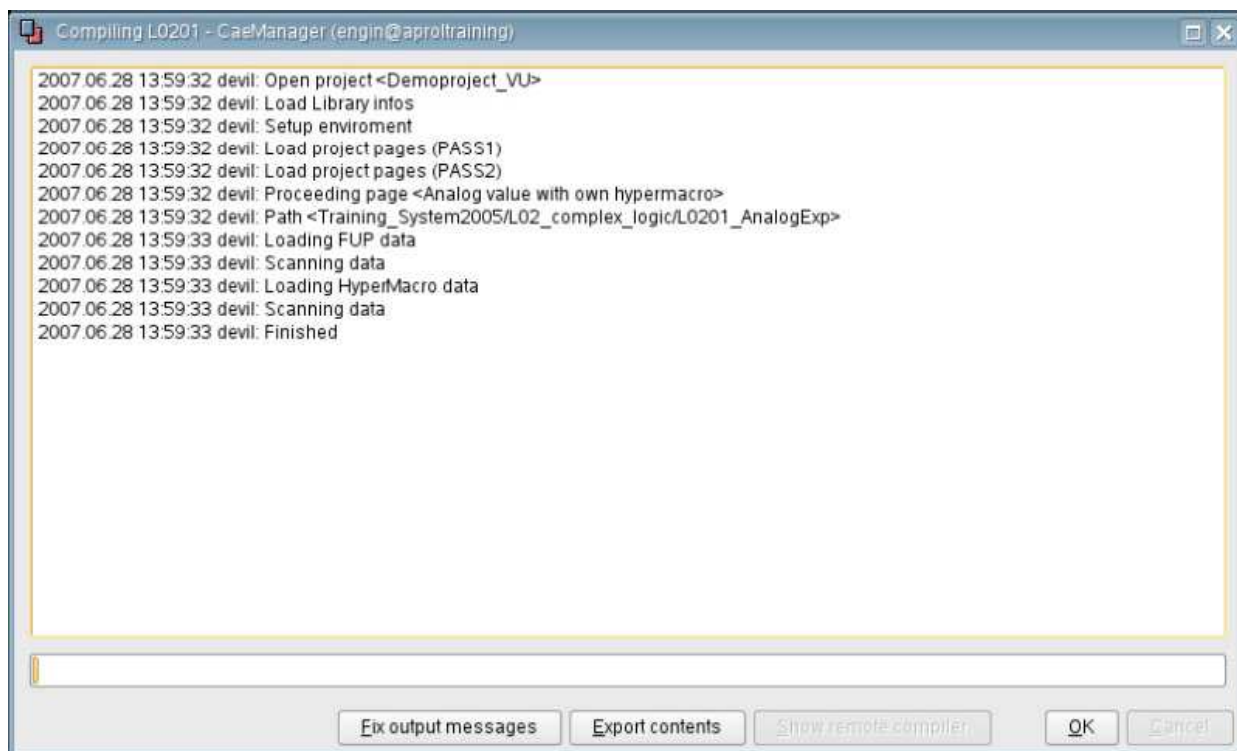


Рис. 68. Стандартная выходная информация при компиляции

Упражнение 5

Вставьте новую структурную схему в созданный каталог и дайте ей логическое имя.

Назначьте ей задачу (помните, что задача должна быть уникальной и соответствовать соглашениям именования).

Измените размер структурной схемы и ознакомьтесь с различными возможностями навигации.

Упражнение 6

Реализуйте мигающий свет, зажигающийся в зависимости от цифрового входа.

Место для подробного изложения:

9. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В APROL

Каждая структурная схема назначается задаче контроллера или задаче управляющего компьютера посредством дерева физического представления задачи. На следующем рисунке показаны эти взаимосвязи.

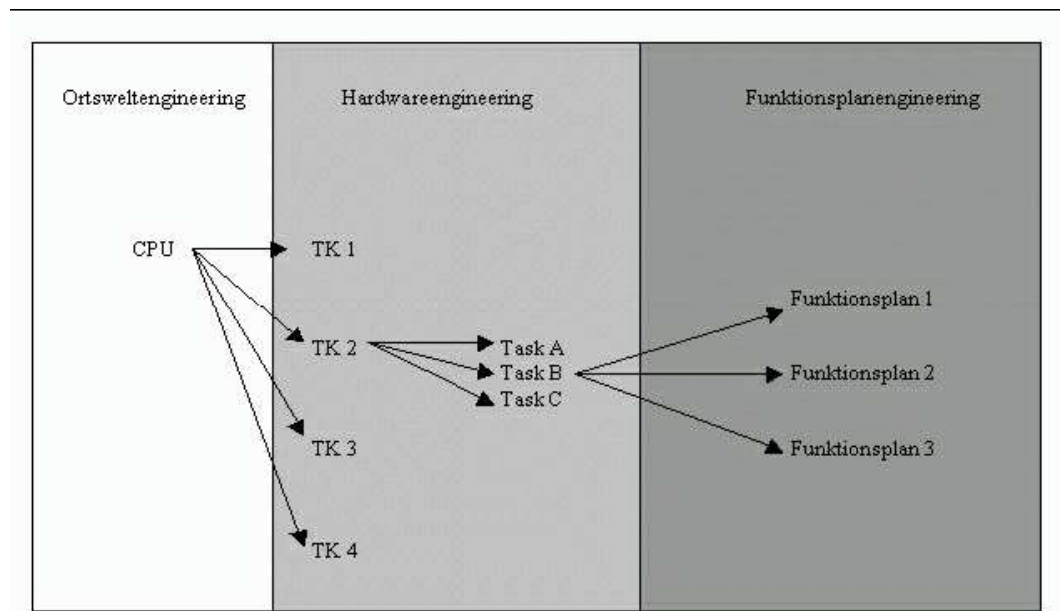


Рис. 69. Диаграмма сгенерированной логики

Определенные технические функции процесса управления соединяются в одно физическое представление. Это означает, что функциональные блоки системы управления могут выполняться только на одном физическом представлении управляющего компьютера. Как правило, логические функциональные блоки хранятся на уровне контроллера. Однако иногда вместо этого удобно выполнять логические функции, такие как предварительный расчет или масштабирование, на уровне управляющего компьютера. Структурирование схем (ориентированных на объекты процессов) не должно искусственно нарушаться в *APROL*. Это позволяет, например, полностью запрограммировать и настроить двигатель в **одной** структурной схеме.

В *APROL* структурные схемы могут обслуживать задачи исключительно процесса управления или контроллера, либо их комбинацию. Если вы создаете задачи исключительно для управления процессом, такие как тренды, сигнализация или настройка визуализации, то в дальнейшем задача управляющего компьютера назначается схеме целиком с помощью дерева физического представления. В этом случае все размещенные функциональные блоки структурной схемы обрабатываются управляющим компьютером. Могут размещаться только логические функциональные блоки, предназначенные для разработки на управляющем компьютере.

Стандартные элементы и гипер-макросы, которые состоят из функциональных блоков контроллера и процесса управления, обычно не могут размещаться на структурной схеме, которой было назначено физическое представление задачи управляющего компьютера.

Примечание. Надежность эксплуатации

В целях надежности эксплуатации не рекомендуется выполнять функции систем, влияющих на надежность, на управляющем компьютере. Этот компьютер не обладает, даже на системах с резервированием, таким уровнем надежности, как контроллер.

Структурные схемы чистых контроллеров включают только функциональные блоки, выполняющие задачи контроллеров. Все функциональные блоки и аппаратные входы/ выходы назначаются выбранным задачам контроллера при назначении физического представления. Подключения аппаратных каналов других систем и структурных схем к схемам в других физических представлениях могут привести к появлению сообщений об ошибках при компиляции и препятствовать активации.

Структурные схемы чистых контроллеров можно с легкостью преобразовать в комбинированные схемы с дополнительными задачами управления процессами. Функциональные блоки, которые должны быть обработаны на управляющем компьютере, объединяются в схему и назначаются посредством дерева физического представления задачи управляющего компьютера. Это означает, что каждый функциональный блок процесса управления должен быть выбран и затем получил запись отдельного физического представления. Функциональные блоки, чье физическое представление не совпадает с физическим представлением на схеме, отмечаются красными квадратами.

Рамки ввода и вывода схем могут также записывать переменные связи и разъемов схем.

Непосредственный обмен данными между двумя контроллерами можно настроить при помощи соединения контроллер-контроллер. Этот тип связи может быть настроен в аппаратном представлении ЦП на вкладке «Communication» (связь).

При компиляции структурной схемы в файловой системе генерируется файл на языке C (экземпляр ИмяСтруктурнойСхемы.C). Если впоследствии генерируется сопутствующая задача, то в файловой системе создается файл с именем «ИмяЗадачи.c». Этот файл включает назначенные вызовы структурной схемы.

10. ГЕНЕРАЦИЯ И РАСПРОСТРАНЕНИЕ

10.1. Версии

Все структурные схемы и другие объекты подпадают под действие системы управления версиями в *APROL*. Управление версиями выполняется автоматически при сохранении структурной схемы. Номера версий увеличиваются автоматически и не управляются вручную. Могут создаваться только новые выпуски (например, от 1.576 до 2.0).

10.2. Активация

В ходе активации вы определяете активную версию, которая далее используется в функции «**Compile**» (компиляция). Активироваться может не только текущая версия объекта. Это часто происходит в случае определения «временных» версий, используемых для тестирования. При этом одновременно с тестированием можно продолжать работать с текущей версией на странице.

При вводе в эксплуатацию следует помнить, что на исполняющей системе всегда используется только активированная версия.

Активация происходит при щелчке правой кнопки мыши на требуемой версии в представлении версий в окне проекта.

Для перехода на представление «**Versions**» (версии) используется контекстное меню объекта.

	Copy	Ctrl+C
	Paste	Ctrl+V
	Rename	
	Delete	Ctrl+D
	Check	F5
	Create version	F6
	Activate	F7
	Compile	F8
	Check (active version)	Shift+F5
	Compile (active version)	Shift+F8
	Generate	F9
	Deactivate	Shift+F7
	Compile/Generate all	Shift+F9
	Set bookmark	F11
	Remove bookmark	Shift+F11
	Lock	Shift+L
	Unlock	Shift+U
	Unlock foreign	Ctrl+Shift+U
	Change Control Logging	Ctrl+L
	References	Shift+F12
	CAE messages	Ctrl+M
	Remove CAE advices	Ctrl+H
	Version view	Shift+V
	Properties	F12

Рис. 70. Открытие представления версий

В рабочей области представления версий в окне проводника отображаются все версии выбранного объекта.

T	LBM	State	Version	C	Instance	Aktivated/Compiled	Created	Version comment
		✓✓	V1.1		L00_FL	06/22/2007 AM09:58:20 CEST aprol	05/23/2007 PM02:40:25 CEST aprol	
			V1.0		L00_FL		12/18/2006 PM04:10:44 CET aprol	test

Рис. 71.
Представление версий

10.3. Компиляция

Компиляция выполняется щелчком правой кнопки мыши на активированной версии страницы проекта в стандартном представлении или представлении версий. При компиляции выполняются различные действия над различными типами страниц проекта. Все эти действия перечисляются под общим заголовком «compilation» (компиляция).

Типы действий, а также информация, необходимая для них, требуют выполнения компиляции объектов в определенном порядке.

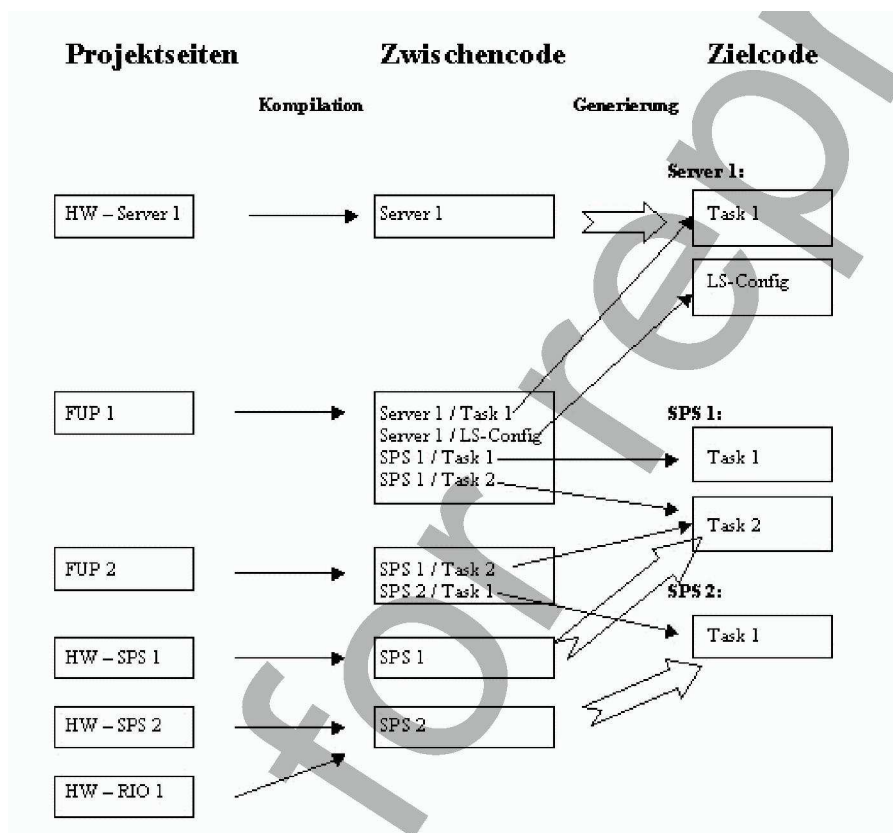


Рис. 72. Процедура и зависимости в ходе компиляции

10.3.1. Компиляция аппаратного обеспечения

При компиляции аппаратного обеспечения:

- Подходящие друг другу объекты комплектуются совместно (например, основные станции вместе со станциями ETHERNET Powerlink).
- Устанавливаются адреса ввода / вывода контроллеров для заданных каналов.
- Выполняется проверка на достоверность для подключенных аппаратных модулей, источника питания и назначений физического представления.

Т.к. данные собираются из отдельных стоек основной станции и всех станций Powerlink, которые также могут подключаться к основной станции, все станции Powerlink, используемые основной станцией, должны быть полностью сконфигурированы, активированы и скомпилированы на данном этапе. При этом внесение любых изменения в станции Powerlink означает необходимость повторной компиляции основной станции.

10.3.2. Компиляция структурной схемы

При компиляции структурных схем язык их программирования (язык функциональных блоков) транслируется в читаемый код (код на языке C) и выделяется в программные модули для различных целевых систем (физическое представление). Для этого требуется, чтобы целевые системы были уже известны, т.е. уже были скомпилированными. Помимо этого, должны быть скомпилированы все библиотеки, в которых структурная схема получает свои функциональные блоки.

Примечание. Генерация кода

Временный код хранится в структуре каталогов проекта. См. курс TM890 Основные сведения о Linux.

10.3.3. Компиляция диаграмм процессов

Диаграммы процессов также относятся к проверке достоверности и выпускаются для компиляции.

10.4. Подтверждение

Помимо различных промежуточных версий страницы проекта, часто существуют более старые, но еще работоспособные версии. Их можно соответственно идентифицировать при помощи функции «**Confirm**» (подтвердить) в представлении версий страницы. Данная функция позволяет перейти к более старой (и, возможно, с отличающейся функциональностью) версии, в которой отсутствуют ошибки. При использовании систем, требующих сокращения (например, FDA), вы можете продолжать работать только с системой с подтвержденными после сокращения версиями.

10.5. Генерация

В зависимости от сгенерированного физического представления выполняются различные действия.

Для контроллера программируемые модули, получаемые в результате компиляции структурной схемы, объединяются с временным кодом и связываются с задачами. На данном этапе эти задачи находятся в состоянии, позволяющем загрузить их в контроллер. Управляющий компьютер должен различать следующие типы конфигурации:

- Логическое содержание структурной схемы следует такому же правилу, которое описано для контроллеров.
- Настройки функционального блока системы управления процессом разделяются (система трендов, система сообщений и т.д.) и хранятся в базе данных.
- Сведения о необходимой взаимосвязи между управляющим компьютером и контроллером размещаются в соответствующих настройках драйвера.
- Диаграммы процессов и динамическое содержимое диаграмм блока картинок, назначенных им, объединяются вместе и снабжаются соответствующим вводом / выводом.

Вся информация объединяется для соответствующего физического представления и готова к загрузке в физическое представление данной формы.

Генерация производится при щелчке правой кнопки мыши управляющего компьютера или основных станций контроллера в представлении проводника.

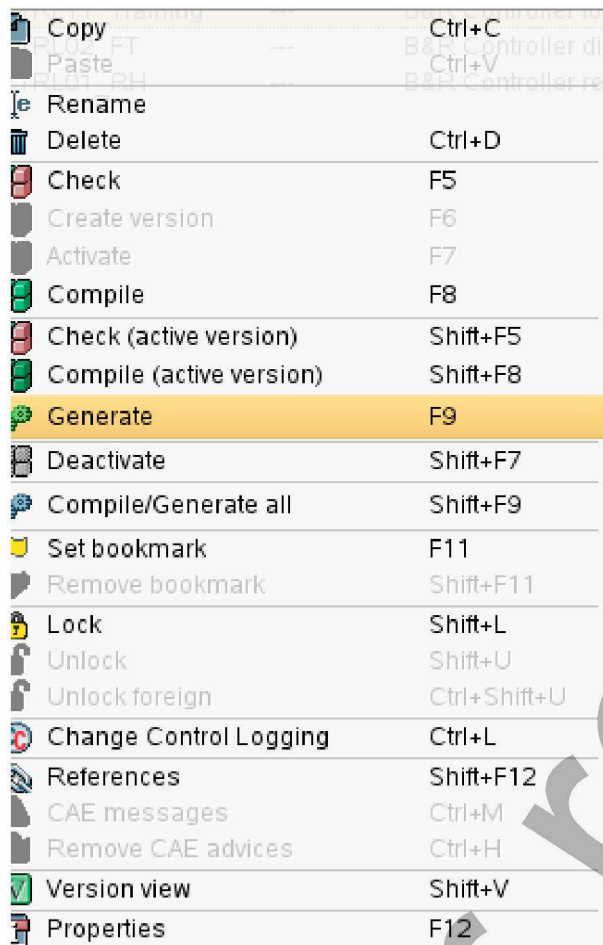


Рис. 73. Команда генерации

В окне «Generate project» (генерация проекта), показанном ниже, отображается список задач, определенных для данного объекта. Список генерируемых задач комплектуется при помощи клавиш со стрелками. Для запуска генерации нажмите «OK».



Рис. 74. Генерация выбора задач

В окне перечислены все сообщения генерации. Сообщения об ошибках выделяются красным цветом. После завершения генерации можно нажать кнопку «ОК», чтобы закрыть окно.

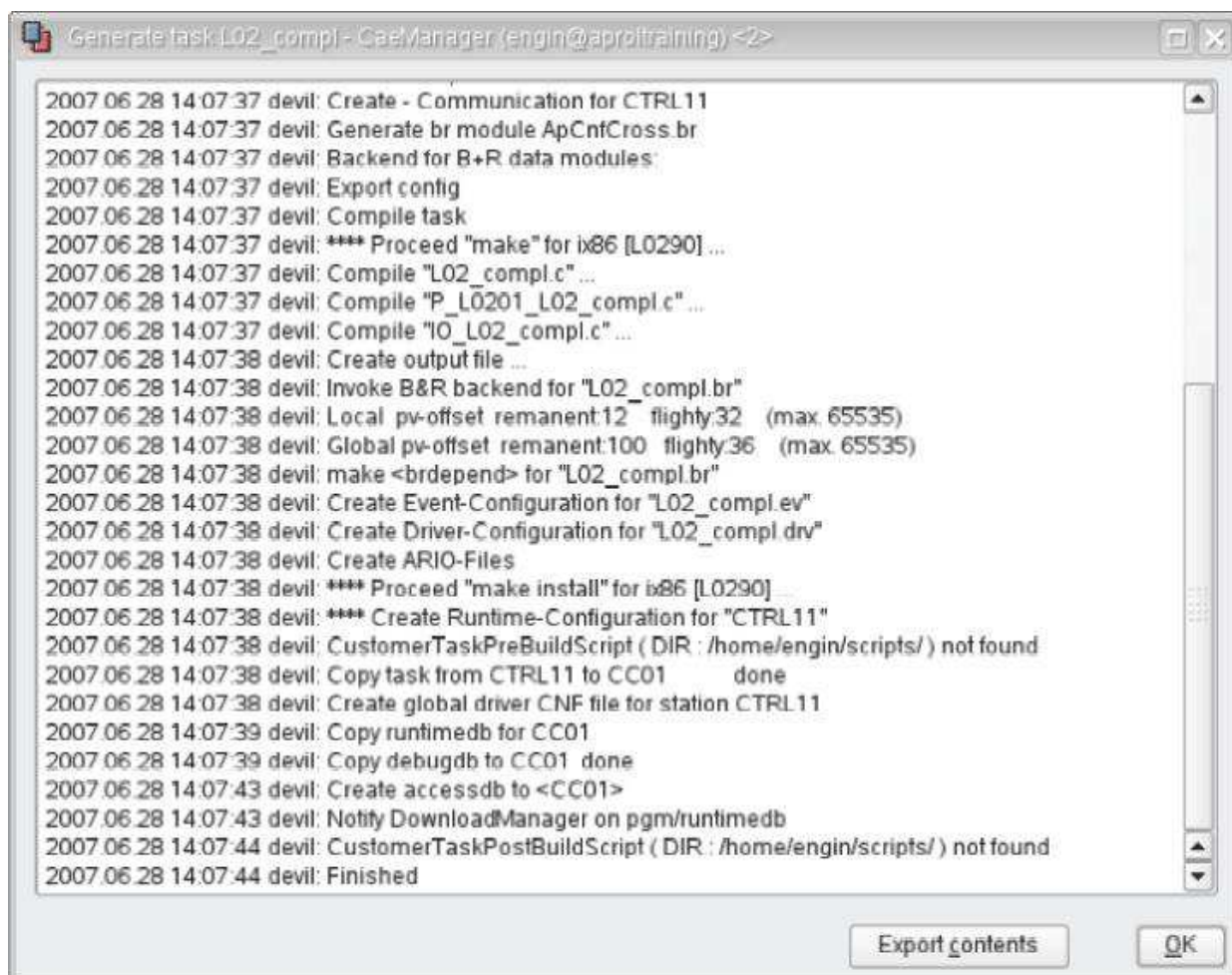


Рис. 75. Вывод компилятора при генерации задачи

10.6. Пункт меню «Повторная компиляция»

Пункт меню **Recompile** (повторная компиляция) находится в главном меню под «**Edit / Recompile**» (редактирование / повторная компиляция) в CaeManager. Если проект открыт, то вы можете выбрать этот пункт, программа запросит способ выполнения компиляции.

Примечание. Повторная компиляция

Повторная компиляция используется при редактировании проекта для проверки выполнения компиляции и генерации всех необходимых объектов. Система *APROL* также сканирует стенд разработки и компилирует только необходимые части проекта.

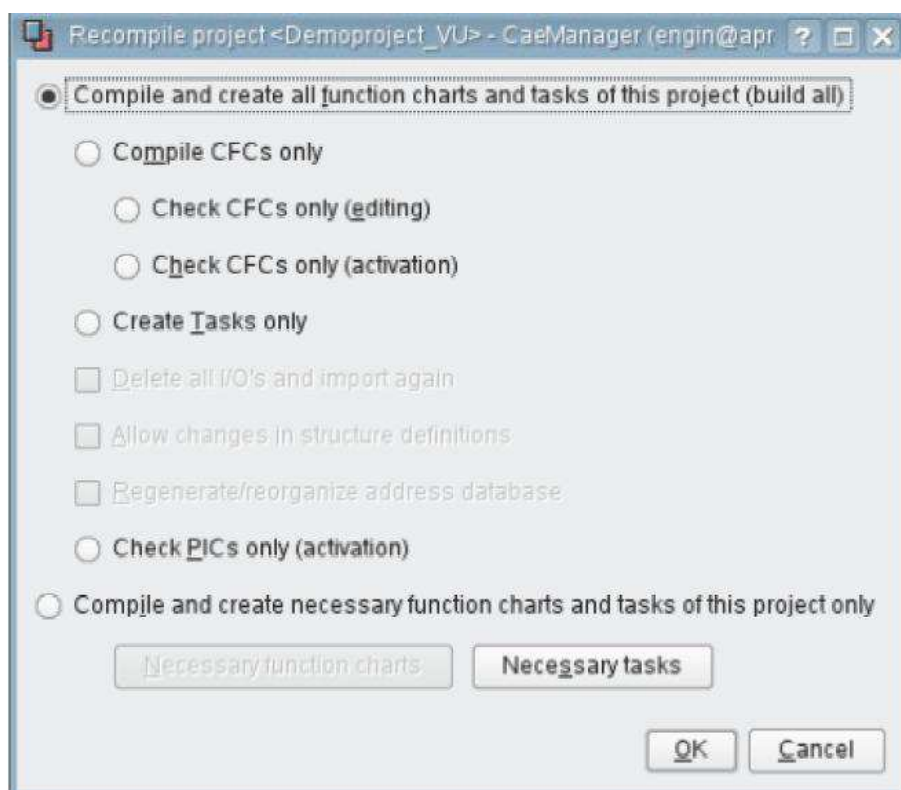


Рис. 76. Выбор возможностей компиляции и генерации

В данном примере компилируются все активированные страницы проекта, генерируются задачи для контроллеров и (при наличии) управляющих компьютеров. После этого проект готов к загрузке в другие системы.

Вы можете изменить параметры так, чтобы страницы не компилировались или не создавались задачи.

Примечание . Генерация

Процесс генерации очень больших проектов может занимать некоторое время!

Данный пункт меню следует использовать только в следующих случаях:

- Вы потеряли трассировку проекта, и требуется много времени на определение планов и страниц, которые необходимо скомпилировать.
- Система *APROL* была изменена.
- Использовался пункт меню **«Compile and create changed function charts and task of this project only»** (компилировать и создавать измененные структурные схемы и задачи только данного проекта), при котором включаются только вновь созданные объекты. Это означает, что система автоматически компилирует необходимые объекты и задачи. Использование указанного пункта меню предоставляет разработчику приложений максимально возможный уровень поддержки и означает, что не требуется отслеживать модули, которые необходимо компилировать после внесения изменений.

На него ссылаются как на «необходимый список».

10.7. Загрузка

После того, как все относящиеся к проекту страницы скомпилированы и все необходимые для проекта задачи сгенерированы, созданные данные необходимо передать в соответствующую целевую систему (контроллер, исполняющую систему и систему оператора). Для каждой целевой системы данные различаются. **Загрузка** – это обобщающий термин для передачи данных в *APROL* (загрузка сгенерированных данных проекта из системы разработки «вниз» в целевые системы). Загрузка выполняется программой **DownloadManager** (диспетчер загрузки).

10.7.1. Общая информация о загрузке

Следующие данные передаются в соответствующие целевые системы:

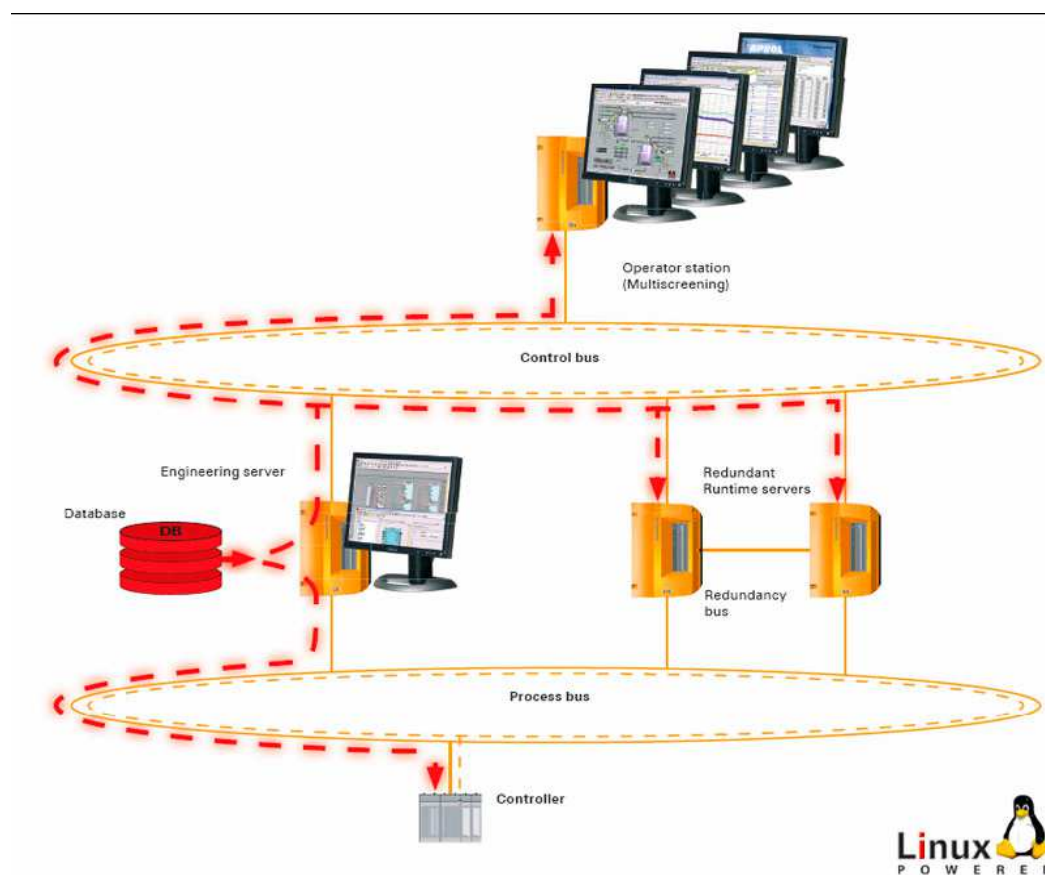


Рис. 77. Схема загрузки

- В целевой контроллер

В контроллер могут быть загружены задачи, библиотеки B&R, пользовательские библиотеки и конфигурация системы для ЦП контроллера. Задачи включают структуры ввода / вывода (соединения между аппаратным вводом / выводом и PV).

- В целевую исполняющую систему (управляющий компьютер)

Управляющему компьютеру предоставляется следующая информация (файлы):

Информация	Комментарий
Модули системы управления	Программы <i>APROL</i> , которые необходимо выполнить на управляющем компьютере в исполняющей системе
Задача управляющего компьютера	Задача (задачи) (сгенерированные из схем для управляющего компьютера), которые должны быть созданы в среде выполнения Система для работы и мониторинга
Настройки	Настройки сигнализации, сообщений, данных трендов и системы сбора данных Chronolog (конфигурация берется из схемы функциональных блоков соответствующей системы управления)
Управление пользователями	Информация, относящаяся к управлению пользователями (права доступа для работы и мониторинга, способы входа в систему)

- В целевую систему оператора (станция оператора)

Станции оператора предоставляется следующая информация (файлы):

Информация	Комментарий
Приложения пользовательского интерфейса	Программы, предоставляемые для работы и мониторинга на станции
Диаграммы процессов	Визуальный интерфейс процесса
Управление пользователями	Информация, относящаяся к управлению пользователями (права доступа для работы и мониторинга, способы входа в систему)

10.7.2. Загрузка после изменений

Влияние изменений при разработке проекта может значительно отличаться и повлечь выполнение различных процедур загрузки.

После внесения изменений в логическое содержание структурной схемы (и, следовательно, задачи) достаточно выполнения простой загрузки задачи на контроллер. Однако, если изменение привело к изменению поведения взаимодействия задачи, то загрузки на контроллер недостаточно. Изменение схемы ввода / вывода контроллера всегда требует перенастройки драйвера шины обработки данных.

В следующих таблицах показаны зависимости изменений конфигурации:

Изменения конфигурации в структурной схеме

Изменение	Необходимая загрузка
Внутренняя логика контроллера или LS	Данная задача
Вводы / выходы структурной схемы	Данная задача
Физическое представление структурной схемы	Старая и новая задачи (возможно, также задача управляющего компьютера)
Физическое представление функционального блока структурной схемы	Старая и новая задачи
Имя экземпляра структурной схемы	Данная задача
Перемещение блоков	Загрузка не требуется

Изменения конфигурации диаграмм процессов

Изменение	Необходимая загрузка
Статичный фон диаграммы процесса	Станция операторов
Анимация диаграммы процесса (без новых блоков картинок в структурной схеме)	Станция операторов
Физическое представление диаграммы	Загрузка не требуется
Имя экземпляра диаграммы процесса	Станция операторов

Изменения конфигурации библиотек

Изменение	Необходимая загрузка
Программная часть функционального блока	Все задачи, в которых используется функциональный блок
Схема ввода / вывода функционального блока	Целиком контроллер, сервер среды выполнения и станция оператора

Изменения конфигурации аппаратного обеспечения

Изменения контроллера	Необходимая загрузка
Новые входы / выходы на старых платах	Загрузка не требуется
Изменения входа / выхода на старой плате	Целиком контроллер, сервер среды выполнения (если ввод / вывод используется на сервере) и станция оператора (если ввод / вывод используется в диаграммах процессов)

Присоединение новой платы	Загрузка не требуется
Вставка новой платы	Целиком контроллер, сервер среды выполнения (если ввод / вывод используется на сервере) и станция оператора (если ввод / вывод используется в диаграммах процессов)
Адресация RIO	Целиком контроллер
Базовые адреса для расширений	Целиком контроллер
Адрес Profibus на плате NW100	Перенастройка Profibus, перезапуск драйвера
Физическое представление аппаратного обеспечения	Целиком контроллер, сервер среды выполнения

Изменения в компонентах системы управления	Необходимая загрузка
Сетевые адреса системы управления процессами	Сервер среды выполнения
Конфигурация модуля системы управления процессами	Сервер среды выполнения
Физическое представление LS	Сервер среды выполнения

Примечание. Загрузка

Вышеприведенные таблицы относятся только к необходимым процедурам загрузки. Следует иметь в виду, что эти изменения обычно требуют значительного объема работы в системе разработки (компиляция, генерация).

Примечание. Загрузка в ходе работы

Для предотвращения риска и обеспечения бесперебойной работы системы необходимо тщательно продумывать любой тип загрузки при работающей системе.

11. ДИСПЕТЧЕР ЗАГРУЗКИ

DownloadManager (диспетчер загрузки) используется для загрузки разработки *APROL* на целевую систему. Запускается из главного меню диспетчера Windows.

В основном существует 4 различных типа загрузки:

- **Загрузка оператора**

Это загрузка на станции операторов. В этом случае загружаются все визуальные элементы, интерактивное управление пользователями и (если выбрано) меню KDE для рабочей среды операторов.

- **Загрузка контроллера:**

Это загрузка скомпилированных задач и библиотек (файлов *.br) на соответствующие контроллеры.

- **Загрузка среды выполнения:**

Это загрузка в среду выполнения из *APROL*. В ходе загрузки все фоновые процессы, такие как задача управляющего компьютера, InaDriver, AlarmServer, TrendServer и т.д., загружаются в новой конфигурации.

- **Загрузка шлюза:**

Это загрузка в систему шлюза. Обычно выполняется при подключении к внешним системам.

В других случаях источник загрузки может отличаться. Процесс загрузки иллюстрируется на следующем рисунке.

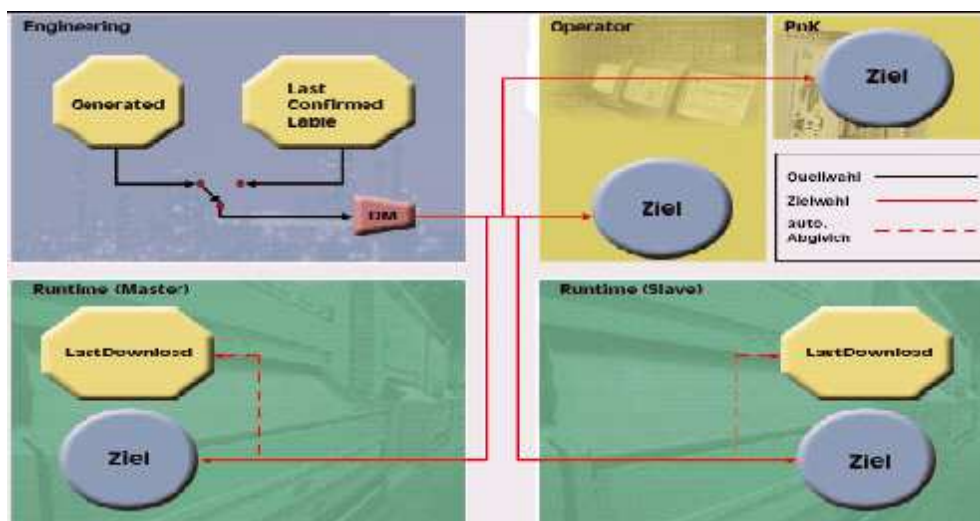


Рис. 78. Схематическое изображение возможностей загрузки

Параметры загрузки (для различных целевых систем (контроллер > IP-адрес, станция оператора > компьютер и целевая система (пользователь)) хранятся в разделе разработки проекта CaeManager и доступны для DownloadManager посредством генерации задачи. При необходимости эти параметры временно могут быть изменены. Это выполняется при помощи «Object manager» (диспетчера объектов), который открывается из контекстного меню целевого объекта загрузки.

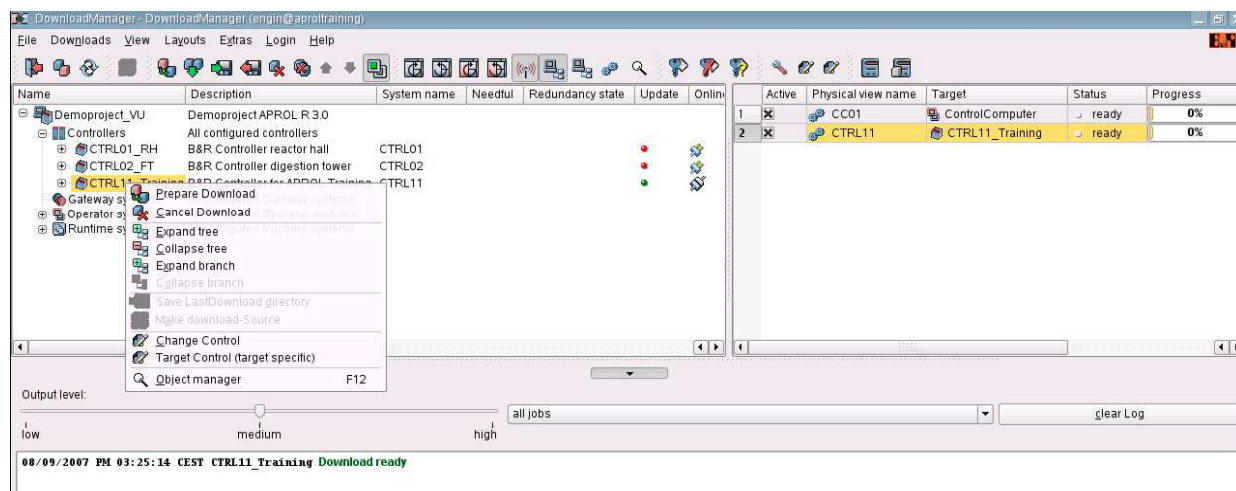


Рис. 79. Окно запуска DownloadManager

Для подготовки объекта к загрузке щелкните его правой кнопкой мыши. Затем можно открыть проводник объекта для загрузки только одного конкретного элемента (например, только одной задачи контроллера, только меню KDE для станции оператора и т.д.).

Загрузка может отображаться следующим образом:

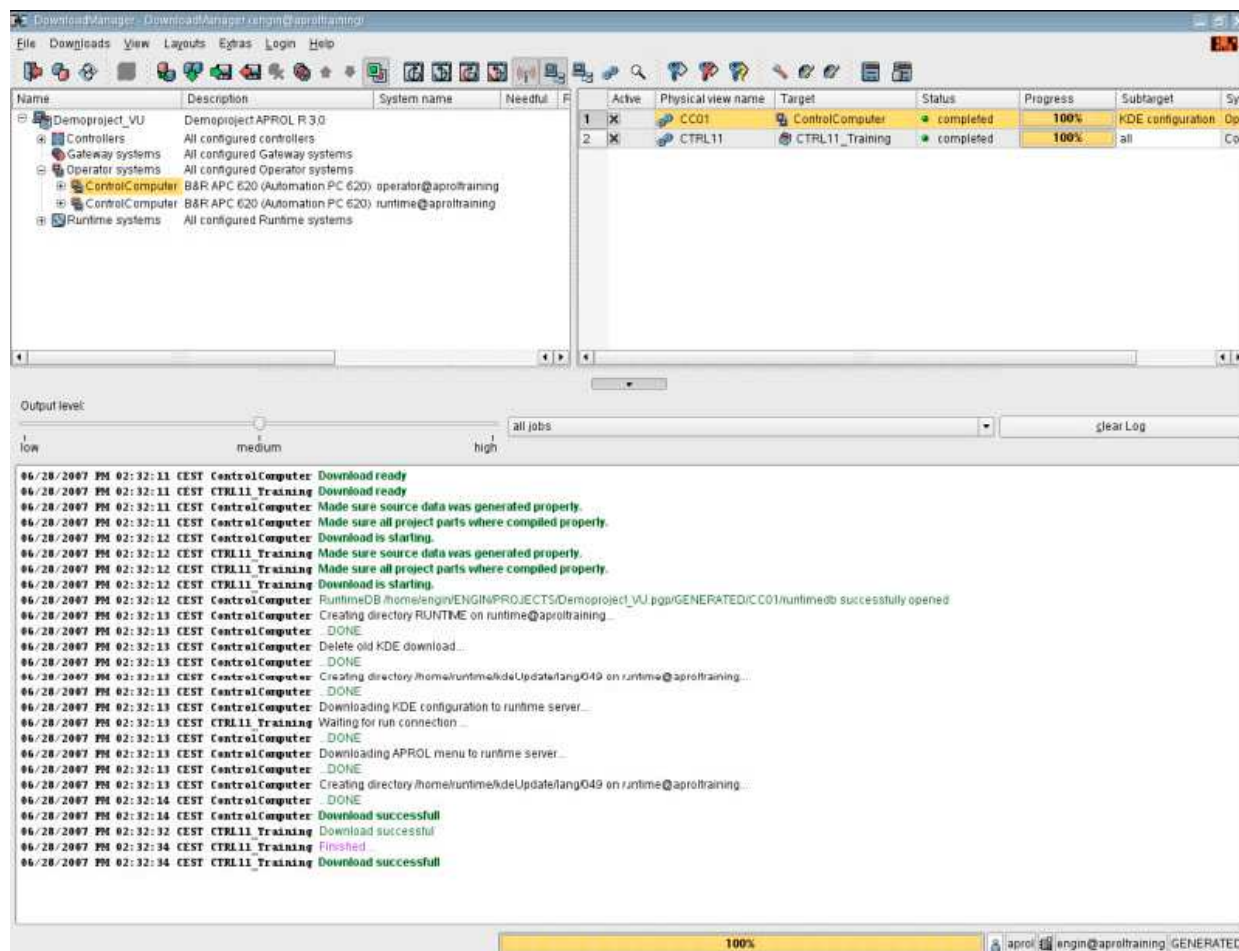


Рис. 80. Выполненная загрузка

Свойства загрузки могут быть изменены для каждого типа загрузки при помощи инспектора объектов. Стандартные параметры, которые отображаются при открытии инспектора объектов, предоставляются и выводятся CaeEngineering.

На следующем рисунке показан инспектор объектов для меню KDE станции оператора:



Рис. 81. Инспектор объектов KDE

Это означает, что тип меню и загружаемые параметры языка для меню могут быть временно изменены для загрузки. Это полезно, если пользователь забыл установить эти параметры в ходе разработки или если в системе оператора временно требуется другое меню.

Инспектор объектов для загрузки контроллера имеет следующий внешний вид:

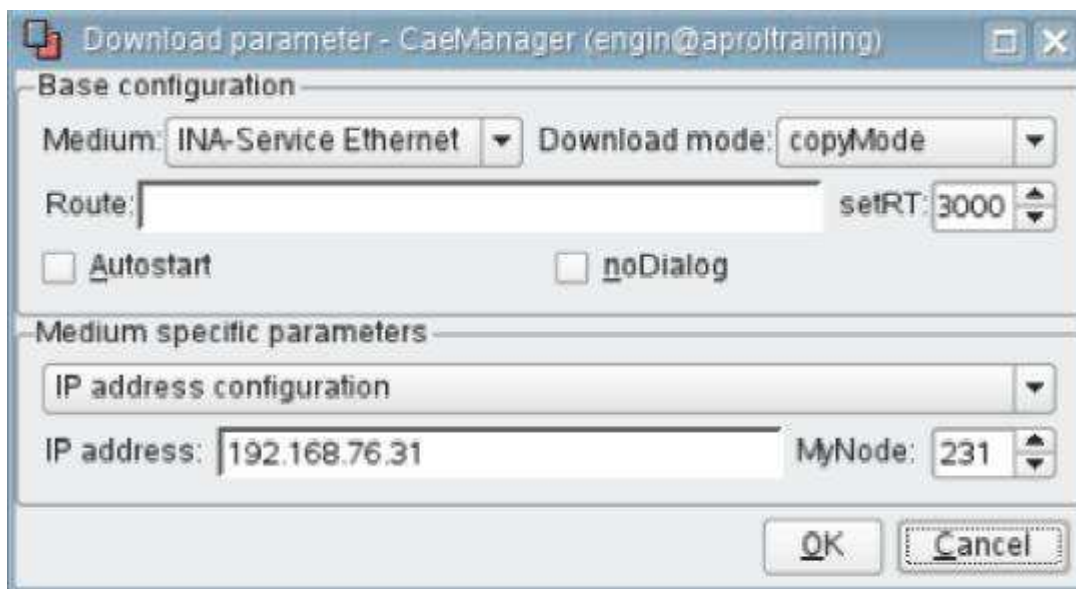


Рис. 82. Инспектор объектов контроллера

В нем можно изменить параметры среды, режим загрузки, IP-адрес и т.д. Это удобно, например, когда Ethernet-подключение к контроллеру еще не установлено. В любом случае загрузка может быть выполнена через последовательное подключение.

Примечание. Сообщения об ошибках

Необходимо обратить внимание на сообщения об ошибках, возникающих в системе.

Подробное описание DownloadManager приведено в справке *APROL*. Также там приводится дополнительная полезная информация. После успешной загрузки система готова к работе. Диспетчер запуска

StartManager (запускаемый из меню KDE) предоставляет информацию о выполняющихся в данный момент процессах, а также отображает их статус. При помощи указанного инструмента допускается вмешиваться в эти процессы.

Упражнение 7

Стенерируйте все необходимые задачи так, чтобы созданная структурная схема работала, а затем разошлите их в соответствующие целевые системы.

Место для подробного изложения:

12. ОТЛАДКА СХЕМЫ

При тестировании до ввода в эксплуатацию часто необходимо отслеживать информационный поток системы более точно, чтобы обнаружить все возможные ошибки конфигурации. С этой целью предоставляется возможность отладки структурной схемы в **CaеManager**. При отладке показываются все разъемы и переменные, которые передаются непосредственно в систему управления процессом в самой схеме. Это означает, в соответствии с текущей версией, что выходы функций, временные переменные и выходы гипер-макросов не могут отображаться или могут отображаться, если схемы скомпилированы с параметром **«Enable debugging of task-local variables»** (разрешить отладку локальных переменных задачи).

Для отладки структурной схемы выберите значок отладки на панели инструментов.

Примечание. Установленная среда

Отладка работает, только если установлена среда выполнения с возможностью запуска. Скомпилируйте и сгенерируйте конфигурацию и выполните загрузку на контроллеры и управляющий компьютер. Также важно, чтобы **CaеManager** был настроен на тот же самый коммуникационный порт (номер порта IOSYS).

Текущие значения отображаются рядом с соответствующей схемой, входами и выходами функционального блока

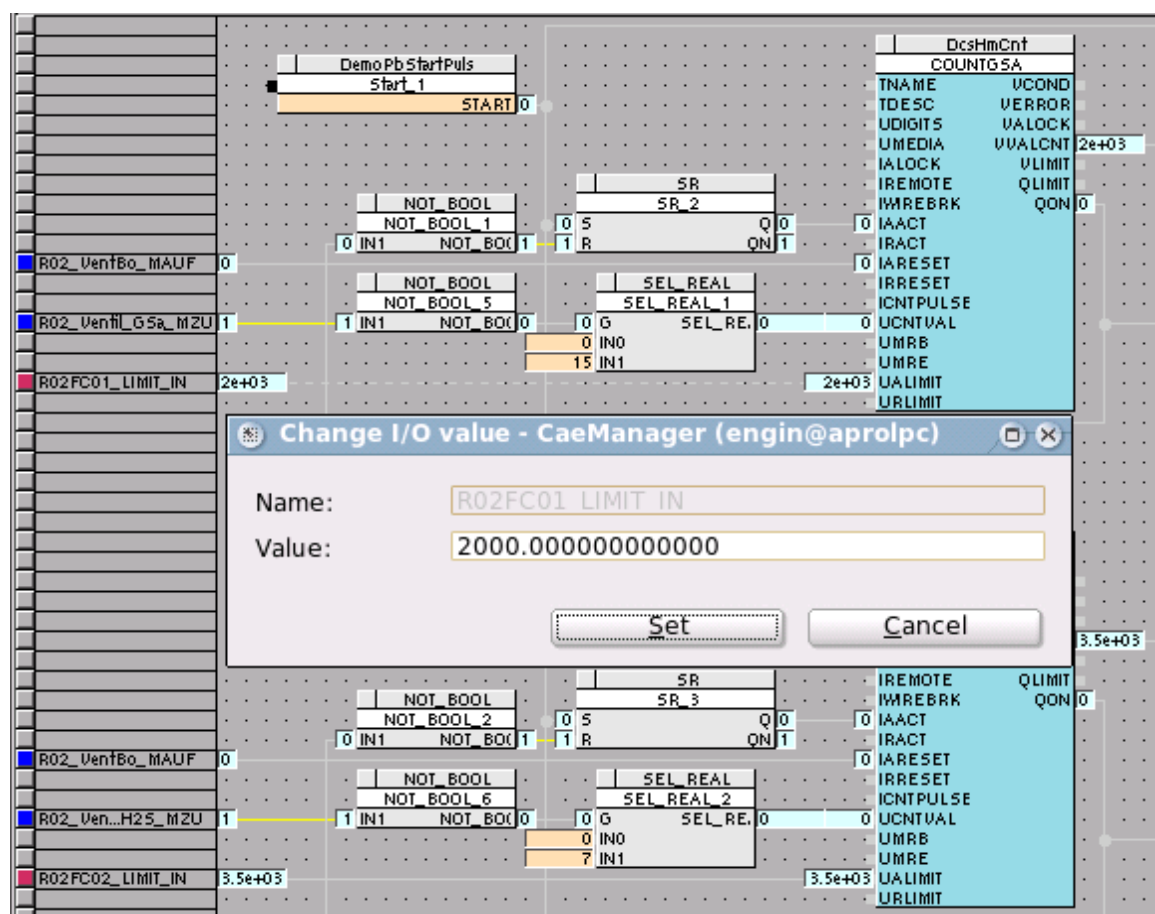


Рис. 83. Отладка структурной схемы

Примечание. Управление памятью

На управляющем компьютере можно определить поведение локальных переменных. Временные переменные, о которых упоминалось выше, также могут отображаться, т.к. управление памятью на компьютерах Linux более гибкое (с подкачкой), и компьютеры обычно также лучше оснащены. Поведение, аналогичное поведению контроллера, возникает при установке флага «- optimize» на управляющем компьютере. Это именно тот случай, при котором регистрируются только переменные, которые соответствуют приведенному примеру. Все остальные переменные хранятся локально в задаче управляющего компьютера и не добавляются к интерактивной базе данных.

12.1. Установка значений

Чтобы установить значение, выберите соответствующую линию питания в структурной схеме при помощи щелчка правой кнопки мыши и используйте функцию «**Change I/O**» (изменить вход / выход). В открывшемся окне вы можете изменить текущее значение. Нажмите кнопку «**Set**» для записи требуемого значения в исполняющую систему. Заметьте, что при установке переменных, отображающих результат вычислений, не всегда показывается требуемый результат. После выполнения одного цикла значение снова будет перезаписано новым результатом вычислений. Оказывает или нет ощутимое влияние установленное вами значение, зависит от разрабатываемой структурной схемы. Полезно использовать эти установки, например, для проверки исключений, позволяющих наблюдать поведение системы при выходе значений за границы. Вы можете даже имитировать работу при невозможных значениях и значениях, не имеющих смысла.

13. ПЕРЕКРЕСТНЫЕ ССЫЛКИ

Перекрестные ссылки упрощают поиск отдельных точек данных в системе, так что их питание / использование может быть проверено быстро и без сложностей. Независимо от этапа разработки вы всегда можете вызвать специальную информацию при помощи пункта меню **«Extras / Cross References»** (дополнительные возможности / перекрестные ссылки). Это позволяет отследить **аппаратные входы / выходы, разъемы, шлюзы ввода / вывода, системные переменные, параметры и константы**.

Помимо этого, пункт меню **«I/O Plausibility»** (достоверность ввода / вывода) позволяет определить, существует ли несогласованность по отношению к переменным (двойное назначение и т.д.).

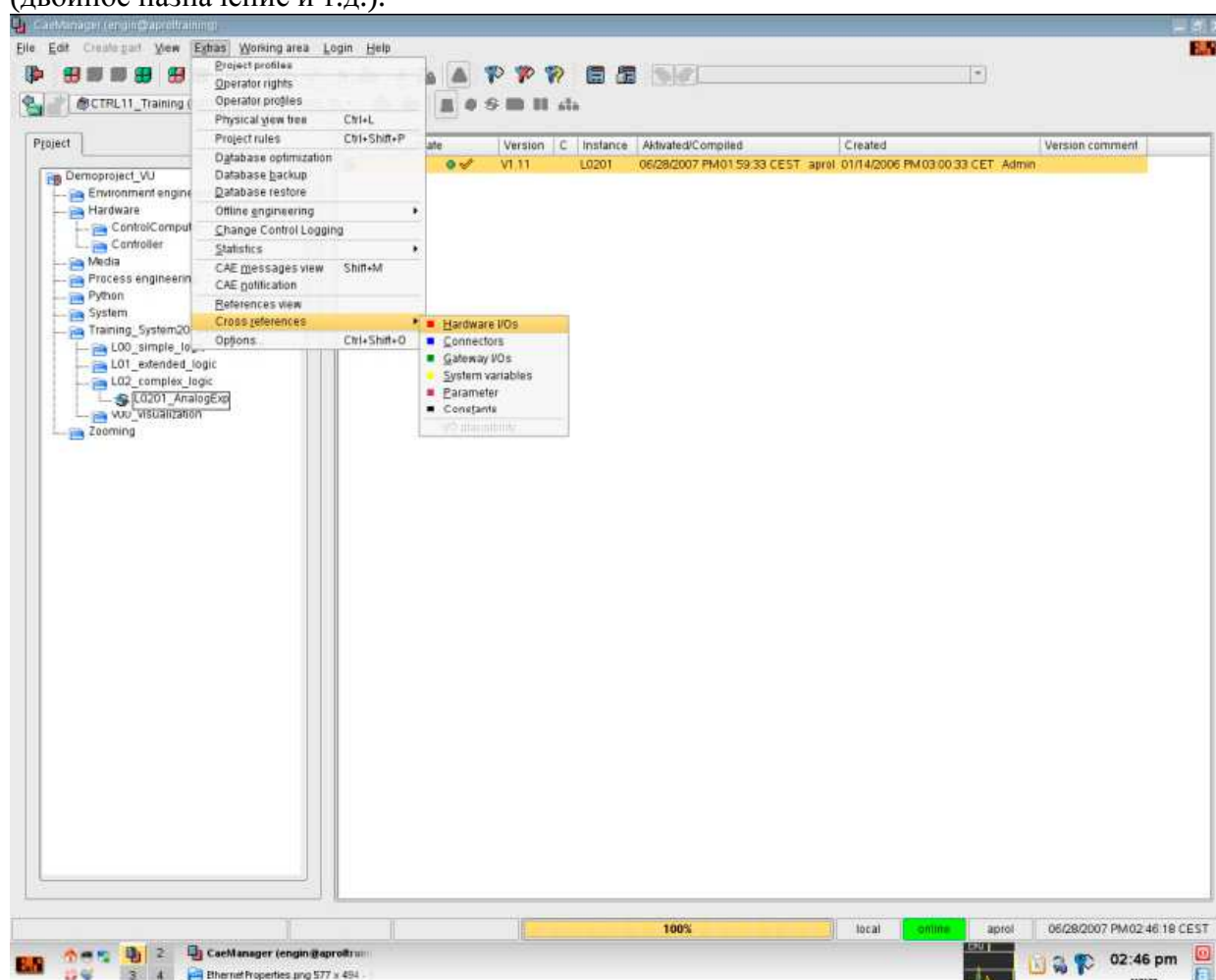


Рис. 84. Открытие перекрестных ссылок

Первое, что вы можете сделать, – отфильтровать требуемую информацию из большого объема данных так, чтобы затем можно было выбрать отдельные точки данных при помощи средней кнопки мыши и открыть их **«Properties»** (свойства).

Hardware I/Os (290/292)

Name	I/O	Event variable	IEC type	Unit	MSR no	Description
Agitator_17201_COn	Ausgang	No	BOOL	---	M17201	agitator of reactor 1 on
Agitator_17201_COn_ST	Eingang	No	BOOL	---	M17201_ST	status message, agitator of reactor 1 on
Agitator_27201_COn	Ausgang	No	BOOL	---	M27201	agitator of reactor 2 on
Agitator_27201_COn_ST	Eingang	No	BOOL	---	M27201_ST	status message, agitator of reactor 2 on
Agitator_27201_Fault	Eingang	No	BOOL	---	M27201_A	reactor 2, agitator bimetal
Agitator_27201_Rpm	Eingang	No	BOOL	---	M27201_SLL	reactor 2, speed sensing switch agitator
Agitator_82105_Fault	Eingang	No	BOOL	---	M82105_A	agitator digestion tower bimetal

Структурная схема может быть открыта щелчком правой кнопки (или двойным щелчком) требуемой записи. Этот прием позволяет избежать навигации по системе разработки проекта.

[illegible]

Рис. 86. Подробное представление перекрестных ссылок с возможностью перехода вперед

14. СОЗДАНИЕ ПРИЛОЖЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЦЕССОМ

Создание приложений систем управления процессами включает визуализацию сигнализации, трендов и других типов накопления данных. Далее это будет проиллюстрировано.

14.1. Создание диаграмм процессов

После того, как разработана структурная схема, следует перейти к диаграммам процессов. Диаграммы процессов формируют интерфейс (HMI) между оператором (техническим персоналом) и системой управления процессом. Диаграммы процессов с оптимальной структурой позволяют пользователю активно воздействовать на настройку и взаимодействовать с выполняющимися процессами. Это выполняется при помощи различных динамических элементов и полей ввода.

Диаграммы процессов создаются в два этапа. Сначала рисуется статичный фон или импортируется уже завершенное изображение, которое будет использоваться в качестве фона. На втором этапе происходит связывание изображения с блоками картинок из структурной схемы, а затем использование их для анимации.

Пункт меню «**Create part / Process diagram**» (создать элемент / диаграмма процесса) используется для создания диаграммы процесса в уже выбранном каталоге. Диаграмме должны быть заданы **имя** и **описание**, а также **экземпляр**.

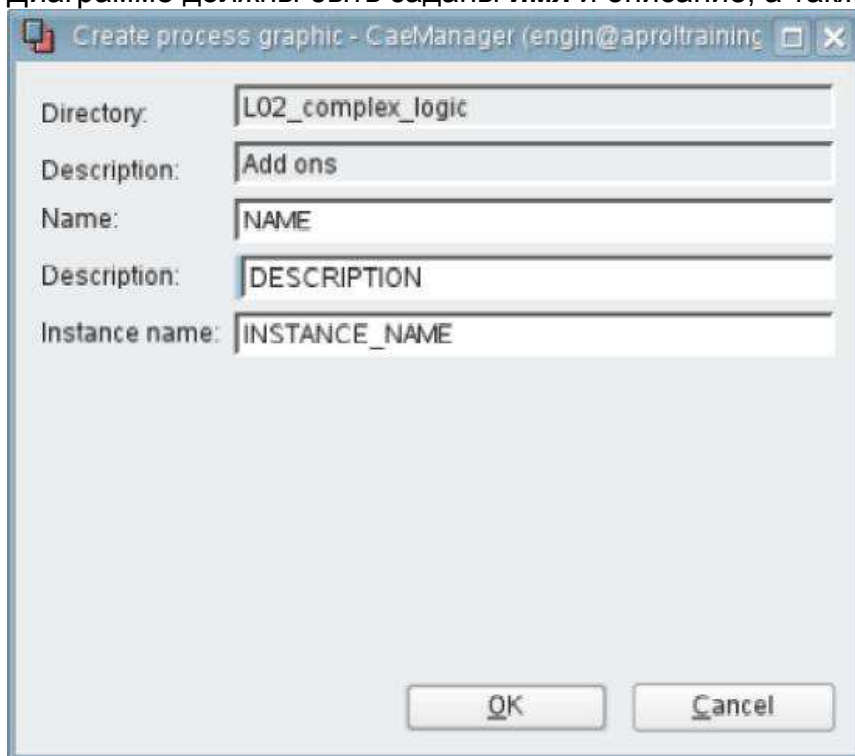


Рис. 87. Создание статичной диаграммы процесса

Примечание . Экземпляр

При задании имен экземплярам для диаграмм процессов необходимо придерживаться соглашения об именах языка программирования C. В противном случае могут возникнуть проблемы при генерации. Экземпляр требуется для таких функций как изменение диаграмм и схема сигнализации (см. справочные файлы для блока AprCcAlarm).

В большинстве случаев элементы статичной картинки (картинки фона) соответствуют диаграммам P&I (схема трубопровода и оборудования). Ограничение на количество используемых диаграмм отсутствует.

В соответствии с нормативами P&I и правилами подразделения технических систем операция должна распределяться по нескольким диаграммам взаимосвязанных процессов (изменение диаграмм посредством вызовов диаграмм (диаграммы с непосредственной связью) → экземпляр диаграммы).

После открытия диаграммы процесса (двойной щелчок объекта) «DisplayEditor» (графический редактор картинок) открывается двойным щелчком предварительного просмотра диаграммы процесса (или щелчком правой кнопки и выбором «Edit» (редактировать)).

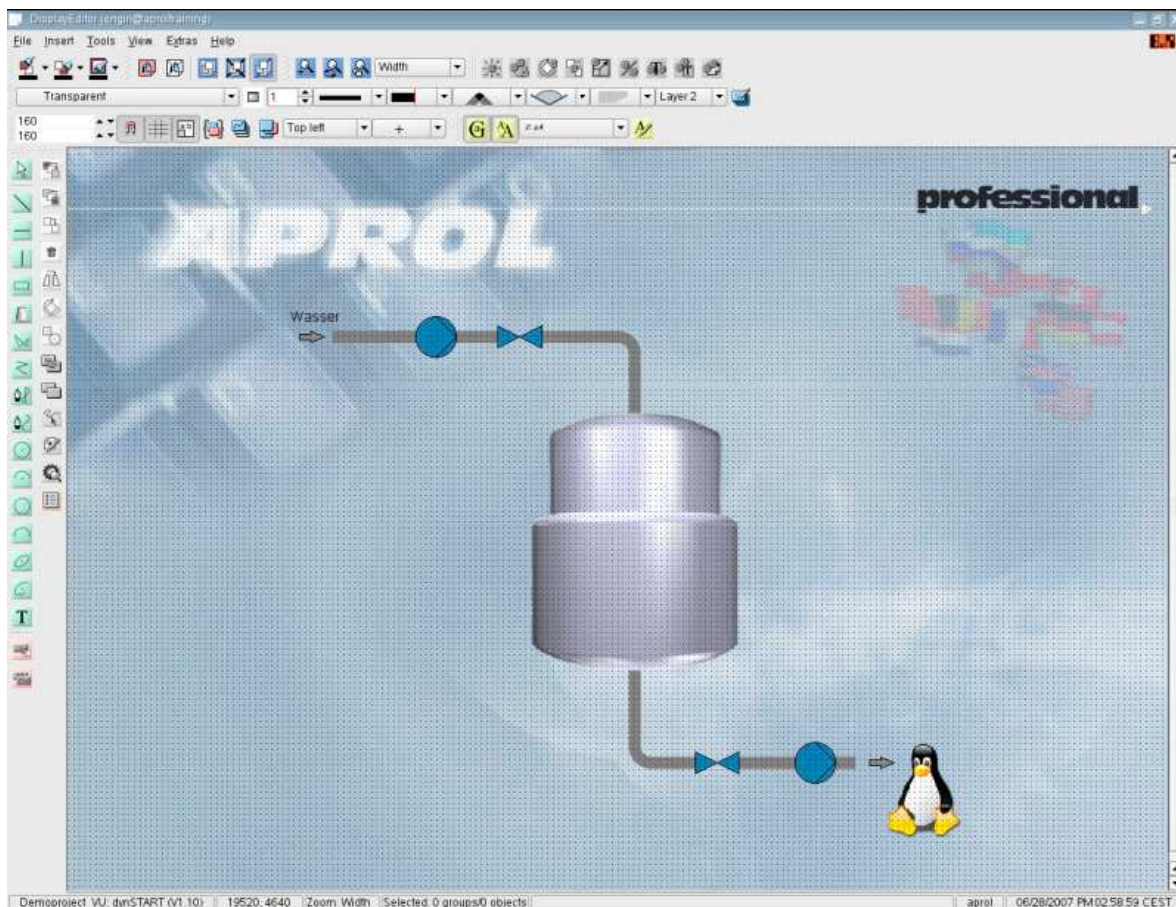


Рис. 88. Экранный редактор

Некоторые функции DisplayEditor:

- Основные функции рисования



Рис. 89. Панель инструментов DisplayEditor

- Работа со слоями

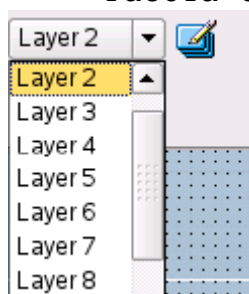


Рис. 90. Выделение слоев

- Многостраничные диаграммы процессов

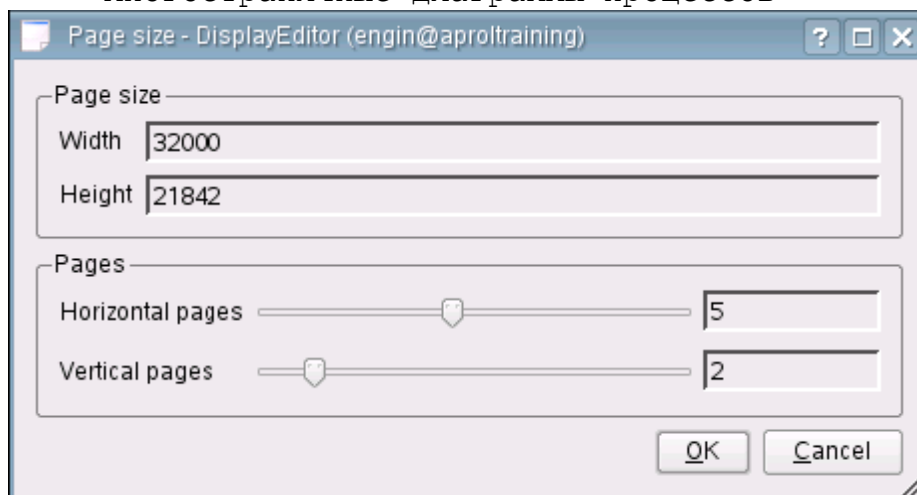


Рис. 91. Настройка многостраничной диаграммы процесса

Многостраничные диаграммы процесса можно реализовать при помощи меню **«File / page size»** (файл / размер страницы).

Это удобно для полного представления системы. Диаграммы процессов в данный момент могут отображаться на нескольких экранах в рабочей среде оператора (многоэкранное отображение).

- Добавление фоновых изображений

DisplayEditor упрощает объединение изображений, созданных различными инструментами (например, CorelDraw и т.д.). Изображения часто определяются на этапе предварительного планирования и, конечно, подготавливаются в графическом виде для предполагаемых и конечных функциональных требований. Эти изображения можно использовать повторно при сохранении в соответствующем формате.



Рис. 92. Добавление фоновых изображений

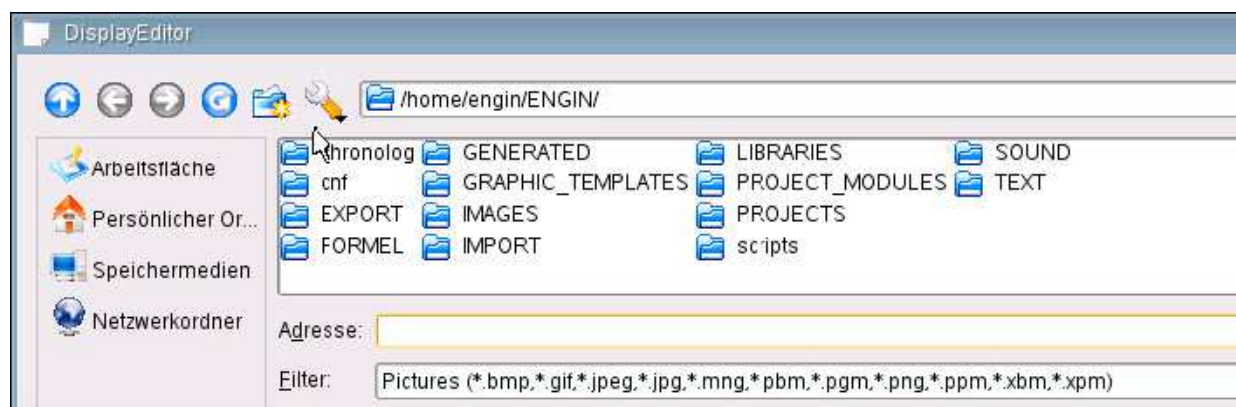


Рис. 93. Возможные форматы

- Добавление графических объектов (например, логотипа и т.д.)

В диаграмме процесса удобно использовать заранее разработанные графические объекты. Допускаются такие же форматы изображений, как и для фоновых изображений.

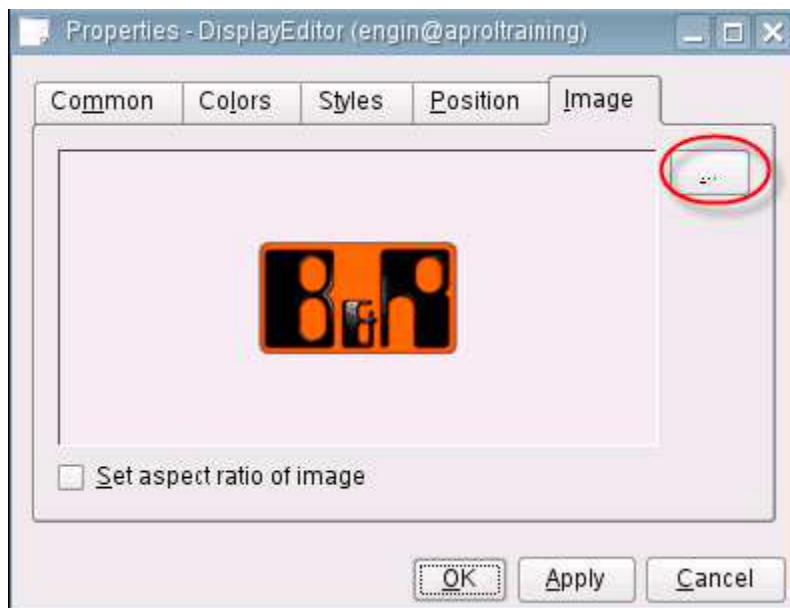


Рис. 94. Добавление графических объектов

- Использование макросов картинок

Макросы картинок, хранящиеся в шаблонах или библиотеках, могут быть повторно добавлены в любое время. Нарисовав картинки один раз, вы можете использовать их многократно.

- Элементы управления и визуализации

Большинство элементов управления и визуализации используется при разработке библиотек. Однако для изменения изображений в других диаграммах процессов требуется, по крайней мере, функция «Picture direct-connect» (картинка с непосредственной связью). Экземпляр диаграммы процесса важен, т.к. должен использоваться и вводиться здесь.

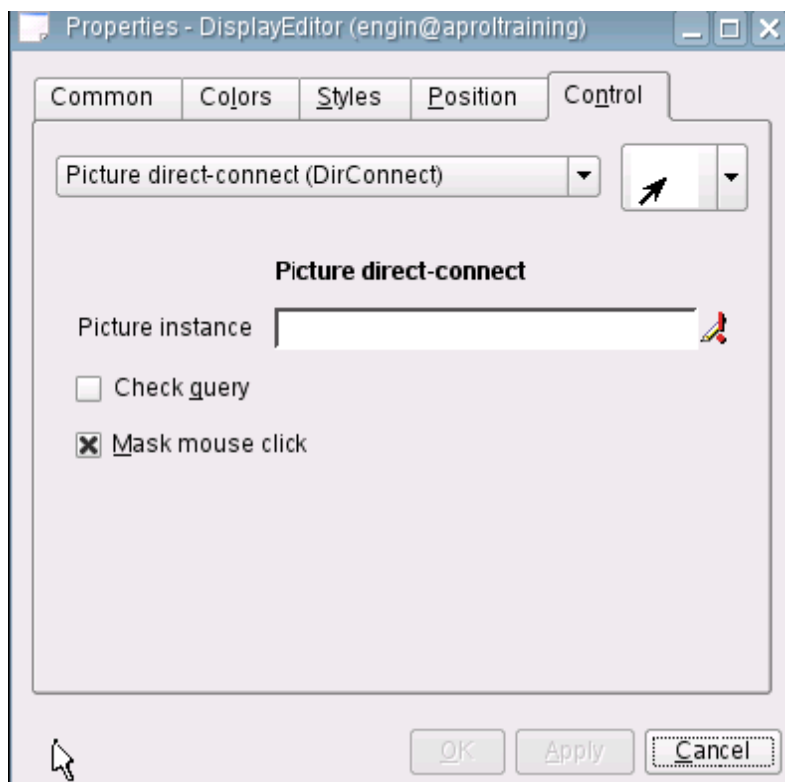


Рис. 95. Вызов изображения

В списке представлены только некоторые функциональные возможности. Как и при использовании других графических программ, вам необходимо ознакомиться с интуитивно понятной работой DisplayEditor.

Для получения подробных сведений см. интерактивную справку *APROL*.

Упражнение 8

Создайте диаграмму процесса. На ваше усмотрение выберите рисование картинки в **DisplayEditor**, объединение различных диаграмм из разных источников или используйте обе возможности. Сгенерируйте задачу управляющего компьютера и загрузите диаграмму процесса в систему оператора при помощи **DownloadManager**. Затем запустите среду выполнения и просмотрите диаграмму процесса.

Место для подробного изложения:

15. ГИПЕР-МАКРОСЫ/ СТАНДАРТНЫЕ ЭЛЕМЕНТЫ

15.1. Что такое гипер-макрос?

Использование САЕ (автоматизированного проектирования) позволяет сохранить значительное время при разработке проекта за счет стандартизации системы или задач. На практике вы пытаетесь привести к общему знаменателю все приводы и точки измерений в системе. Поэтому все установленные устройства с приводами (насосы, мешалки, двигатели, вентиляторы и т.д.) будут управляться по большей части одинаковым способом. Таким образом, совпадает не только ручное управление всех установленных приводов, но и лицевых панелей. Иногда отличаются только условия блокировки.

Точки измерений можно грубо разбить на две группы: аналоговые точки измерений с числовой индикацией (которая всегда может быть реализована одинаково) и двоичные точки измерений, наподобие поплавковых переключателей, которые могут иметь только один индикатор.

За счет однократного создания идентичных функций, доступных в любое время, экономится время проектирования. Для решения подобных задач **APROL** предоставляет гипер-макросы.

Объяснение гипер-макросов:

Гипер-макрос – это структурная схема, которая трансформируется в логический функциональный блок. Функциональные возможности контроллера, управляющего компьютера, динамической визуализации, сигнализации, трендов, трафика шины и т.д. для стандартного приложения (например, установки привода) полностью программируются в этом типе структурной схемы.

На данном этапе отсутствует назначение реально существующему оборудованию ввода / вывода, т.к. функциональный блок должен иметь возможность универсального использования. Заглушками на рамках входа и выхода схемы являются контакты готового функционального блока, которые затем соединяются с соответствующими переменными процесса. В проекте гипер-макрос используется так же, как любой другой функциональный блок, схемы входов и выходов которого только записываются или считываются.

Это приводит к ускорению процесса настройки, т.к. логика для мотора выполняется одним гипер-макросом в структурной схеме. При помощи гипер-макроса подается питание и выполняется настройка, а затем необходимо только сохранить динамическую часть в диаграмме процесса. Это все, что необходимо для разработки стандартных элементов.

Гипер-макросы встраиваются в библиотеки и поэтому могут извлекаться из них для настройки (как объяснялось при разработке структурных схем).

У гипер-макросов есть собственные свойства (вызываемые также средней кнопкой мыши) с несколькими различными параметрами настройки, позволяющими подключать контакты и использовать константы. Переменные используются в качестве констант гипер-макросов (определенных при разработке библиотек) и не могут изменяться интерактивно в среде выполнения.

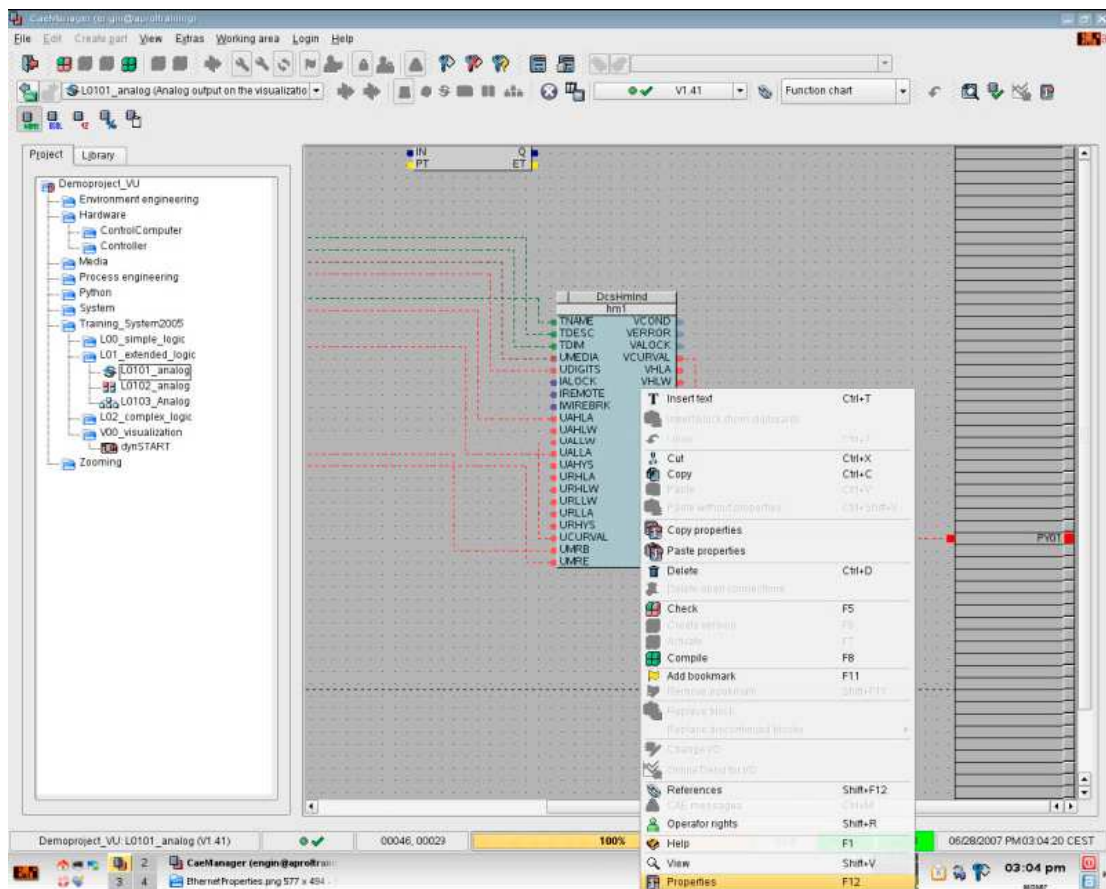


Рис. 96. Открытие свойств гипер-макроса

Записи для констант гипер-макроса в большинстве случаев включают системы трендов и сигнализации, т.к. функции этих систем невозможно настроить интерактивно (см. справку с исключениями).

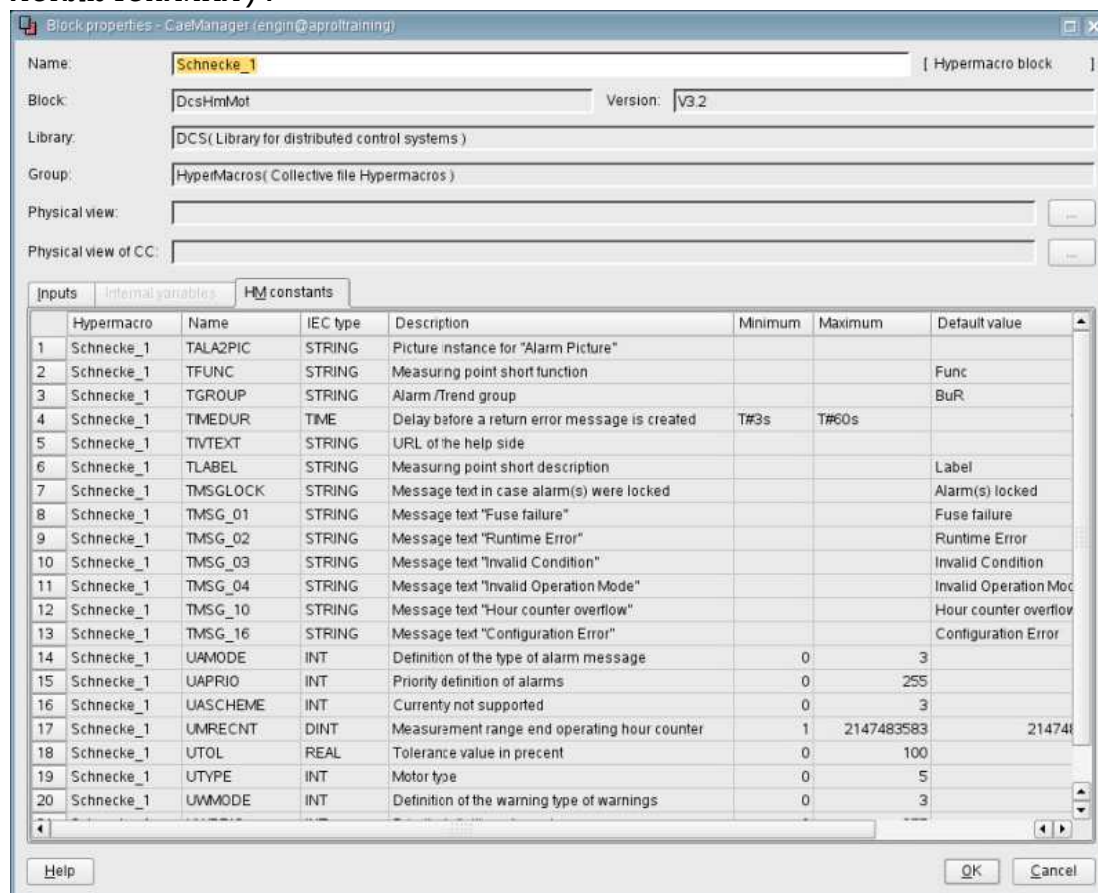


Рис. 97. Свойства гипер-макросов – сведения о константах НМ

Различная функциональность дает возможность использования вложенных гипер-макросов. За счет вложенности для упрощения и ускорения разработки можно реализовать стандартные элементы для неоднократно используемой в проекте функциональности. Переменные, созданные системой, длина которых не может превышать 32 символов в физическом представлении контроллера, ограничивают вложение гипер-макросов.

Примечание. Гипер-макрос в гипер-макросе

При разработке библиотек экземпляры блоков, используемых в гипер-макросе, должны быть по возможности короткими и сжатыми для реализации наиболее глубокого уровня вложения гипер-макросов.

Если гипер-макрос успешно настроен и снабжен параметрами, то не хватает только «связи» между элементами динамической диаграммы структурной схемы и существующей диаграммой процесса. Для установления связи откройте диаграмму процесса и выберите элемент динамической визуализации из структурной схемы во второй вкладке и поместите его на диаграмму процесса (перетащите).

Для точного размещения элемента диаграммы существует сетка, доступная вместе с функцией «snap-to-grid» (фиксация в углах сетки). Все размещаемые элементы могут масштабироваться и вращаться при помощи меню свойств.

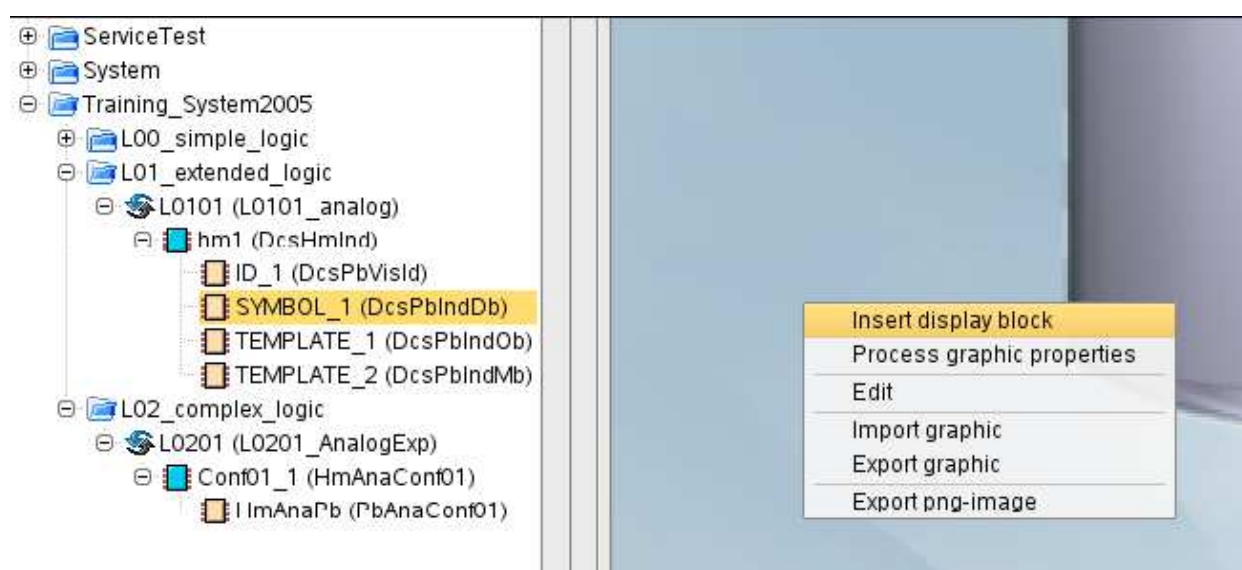


Рис. 98. Связывание логики и диаграммы процесса

Упражнение 10

Разместите гипер-макрос DcsHmInd из библиотеки DCS на созданной вами структурной схеме и настройте его. При выполнении задания обращайтесь к справочной информации по гипер-макросам. Разместите один из макросов внутренних картинок на вашей диаграмме процесса.

Дополнительные действия: Этап создания, этап загрузки, среда выполнения > проверка.

Место для подробного изложения:

16. КРАТКИЕ ВЫВОДЫ

Чего мы достигли к этому моменту?

СaeManager, в качестве основного инструмента проектирования, является ядром среды разработки. Поэтому важно ознакомиться с его структурой. При возникновении каких-либо проблем следует обращаться к справочной системе *APROL*.

Все проектирование выполняется при помощи данного инструмента, позволяющего выполнить работу быстрее и с большей эффективностью. Эти преимущества становятся еще более очевидными, если пользователь полностью понимает все возможности понятия гипер-макросов.

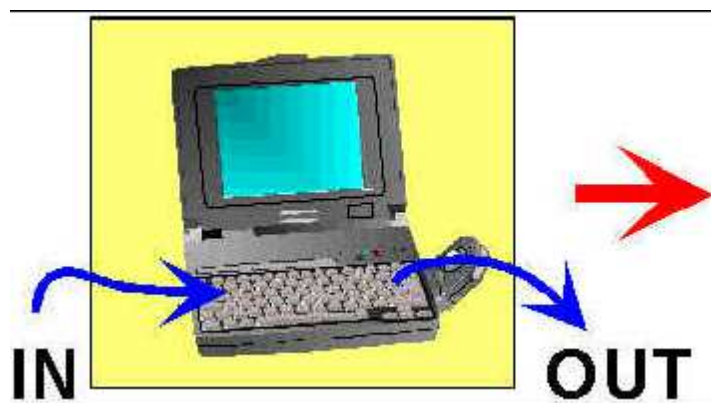


Рис. 99. Все уже сделано – осталось только выключить компьютер

16.1. Список контрольных вопросов (короткие вопросы по разделам)

Какие рабочие области включены в CaeManager?

Какие преимущества разработки аппаратного обеспечения существуют в *APROL*?

Каково назначение физического представления?

Какие существуют целевые системы?

Что такое локальная переменная?

Для чего нужно подключение контроллер-контроллер?

Что такое гипер-макрос?

Что такое динамический элемент?

16.2. Ответы на контрольные вопросы

Обратите внимание на необходимую дополнительную информацию.

Примечания

17. ПРИЛОЖЕНИЕ

17.1. Журнал версий

Версия	Дата	Изменение
V1.00	01.01.2007	Начало журнала
V1.10	01.01.2008	Исправлена в соответствии с преобразованием APROL R3.2 и проверена

Имя файла настоящего документа

APROL_R32_S7_ProjectEngineering_ENG.pdf

Обзор учебных программ

TM200 - Презентация компании B&R **	TM410 - Основные сведения о визуализации
TM201 - Спектр продуктов B&R **	TM410 - Основные сведения об ASiV
TM210 - Основные сведения об Automation Studio	TM630 - Руководство по программированию визуализации
TM211 - Интерактивное взаимодействие Automation Studio	TM640 - Система сигнализации ASiV
TM212 - Объекты автоматизации **	TM650 - Интернализация ASiV
TM213 - Среда выполнения Automation Runtime	TM660 - Удаленные устройства ASiV
TM220 - Работа технического специалиста по обслуживанию	TM670 - Расширенные возможности ASiV
TM221 - Компоненты автоматизации и источники ошибок *	
TM223 - Система диагностики Automation Studio	TM700 - Automation Net PVI
TM230 - Разработка структурированного программного обеспечения	TM710 - Средства связи PVI
TM240 - Многозвенная диаграмма (LAD)	TM711 - Программирование PVI DLL
TM243 - Последовательные функциональные схемы (SFC) *	TM712 - PVI Services
TM245 - Список команд (IL) *	TM730 - PVI OPC
TM246 - Структурированный текст (ST)	
TM247 - Основы автоматизации (AB) *	TM800 - Концепция системы APROL
TM248 - ANSI C	TM801 - Основы проектирования APROL
TM250 - Управление памятью и хранение данных	TM810 - Установка, настройка и восстановление APROL

TM260 - Библиотеки Automation Studio I	TM811 - Система среды выполнения APROL
	TM812 - Диспетчерское управление APROL
TM400 - Основные сведения об управлении перемещениями	TM813 - APROL XML-запросы и след контроля
TM402 - Системы управления перемещениями с привязкой к координатам *	TM830 – Проектирование в APROL
TM410 - Основные сведения об ASiM	TM840 - Управление параметрами и наборы команд APROL
TM440 - Основные функции ASiM	TM850 - Настройка контроллера и INA в APROL
TM441 - Функции ASiM с несколькими координатными перемещениями	TM860 - Разработка библиотек APROL
TM445 - Программное обеспечение ACOPOS ACP10	TM865 - Руководство по библиотекам APROL
TM450 - Концепция управления и настройка ACOPOS	TM870 - Программирование на Python в APROL *
TM460 - Запуск двигателей *	TM880 - Отчеты APROL*
	*) по запросу
	**) см. каталог продуктов

Рисунок	EN	RU
77	Operator station (multiscreening)	Станция оператора (многоэкранная)
	Control bus	Шина управления
	Engineering server	Сервер разработки
	Database	База данных
	Redundant runtime servers	Резервные серверы среды выполнения
	Redundancy bus	Резервная шина
	Process bus	Шина обработки данных
	Controller	Контроллер
78	Engineering	Разработка
	Last confirmed table	Последняя подтвержденная таблица
	Runtime (master)	Среда выполнения (главная станция)
	Runtime (slave)	Среда выполнения (подчиненная станция)
	Operator	Оператор
	Ziel	Целевой объект
	Last download	Последняя загрузка
99	IN	ВХОД
	OUT	ВЫХОД