

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
РАЗДЕЛ 1. ТЕОРЕТИКО-ПРАКТИЧЕСКИЕ АСПЕКТЫ МНЕМОСХЕМ ВИЗУАЛИЗАЦИИ ТЕХНОЛОГИЧЕСКОГО ОБЪЕКТА	7
1.1 Общие принципы разработки мнемосхем при помощи SCADA-систем	7
1.2 Обзор SCADA - систем используемых в производстве	12
1.3 Выбор SCADA - системы для разработки мнемосхемы	19
РАЗДЕЛ 2 РАЗРАБОТКА МНЕМОСХЕМЫ ВИЗУАЛИЗАЦИИ ТЕХНОЛОГИЧЕСКОГО ОБЪЕКТА С ПОМОЩЬЮ SCADA- СИСТЕМЫ	22
2.1 Постановка задачи создания мнемосхемы технологического объекта	25
2.2 Отображение состояний механизмов и датчиков на технологических мнемосхемах	34
2.3 Технологические параметры мнемосхемы	
2.4 Разработка мнемосхемы технологического объекта	39
РАЗДЕЛ 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ХРАНЕНИЯ ДАННЫХ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА	45
3.1. Разработка базы данных средствами СУБД MySQL	45
3.2. Разработка программы для пересылки данных технологических параметров в БД	48
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	54
ПРИЛОЖЕНИЕ А ИЛЛЮСТРАЦИИ ДИПЛОМНОЙ РАБОТЫ	55
ПРИЛОЖЕНИЕ Б СЦЕНАРИИ, ИСПОЛЬЗУЕМЫЕ ПРИ ПОДКЛЮЧЕНИИ БАЗЫ ДАННЫХ	69

						EEEE 000000 000 000 AA		
Изм	Лист	№ докум.	Подпис	Дата	Разработка мнемосхемы визуализации технологического объекта "Повысительная насосная станция с фильтровальной установкой"	Литер	Лист	Листов
Разраб.		Капцов Р.В.	ь			а	3	70
Проверил						ППП-2222 – 111/999		
Т.контр.								
Н.контр.								
Утв.								

ВНИМАНИЕ!

Если ты это читаешь, то наверняка нашёл эту презентацию и попытаешься конвертировать её в Word?

Сразу скажу, это задание по SCADA ЛЕГЧЕ, чем программирование. Нужно всего лишь построить схему из объектов и привязать теги – для того, чтобы при их изменении, объекты меняли цвет. Ну и связать это дело с базой данных.

Не КОПИРУЙТЕ ВСЁ ПОДРЯД!

А так, вроде всё. Удачи там!!! И на защите тоже!!!

ВВЕДЕНИЕ

SCADA — программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, АСКУЭ, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д. SCADA-системы используются во всех отраслях хозяйства, где требуется обеспечивать операторский контроль за технологическими процессами в реальном времени. Данное программное обеспечение устанавливается на компьютеры и, для связи с объектом, использует драйверы ввода-вывода или OPC/DDE серверы. Программный код может быть как написан на одном из языков программирования, так и сгенерирован в среде проектирования.

Иногда SCADA-системы комплектуются дополнительным ПО для программирования промышленных контроллеров. Такие SCADA-системы называются интегрированными и к ним добавляют термин SoftLogic.

Термин «SCADA» имеет двоякое толкование. Наиболее широко распространено понимание SCADA как приложения, то есть программного комплекса, обеспечивающего выполнение указанных функций, а также инструментальных средств для разработки этого программного обеспечения. Однако часто под SCADA-системой подразумевают программно-аппаратный комплекс. Подобное понимание термина SCADA более характерно для раздела телеметрия.

Значение термина SCADA претерпело изменения вместе с развитием технологий автоматизации и управления технологическими процессами. В 80-е годы под SCADA-системами чаще понимали программно-аппаратные комплексы сбора данных в реальном времени. С 90-х годов термин SCADA больше

используется для обозначения только программной части человеко-машинного интерфейса АСУ ТП. [1]

В данной работе необходимо реализовать мнемосхему технологического объекта “Повысительная насосная станция с фильтровальной установкой”. Для этого, необходимо рассмотреть общие принципы разработки мнемосхем при помощи SCADA-систем, произвести обзор SCADA-систем, используемых на производстве и выбрать SCADA-систему для разработки мнемосхемы. Затем сделать постановку задачи создания мнемосхемы технологического объекта, разработать состояние механизмов и датчиков на мнемосхеме, описать технологические параметры мнемосхемы и разработать саму мнемосхему. После разработки мнемосхемы, необходимо будет разработать базу данных средствами СУБД MySQL и программу для пересылки данных технологических параметров в базу данных.

РАЗДЕЛ 1. ТЕОРЕТИКО-ПРАКТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ МНЕМОСХЕМ ВИЗУАЛИЗАЦИИ ТЕХНОЛОГИЧЕСКОГО ОБЪЕКТА

1.1 Общие принципы разработки мнемосхем при помощи SCADA-систем

Мнемосхема — совокупность сигнальных устройств и сигнальных изображений оборудования и внутренних связей контролируемого объекта, размещаемых на диспетчерских пультах, операторских панелях или выполненных на персональном компьютере. [2]

Она представляет собой наглядное графическое изображение функциональной схемы управляемого или контролируемого объекта. Это может быть технологический процесс, энергетическая система, цех станков с числовым программным управлением и т.п. Иначе говоря, мнемосхема - это условная информационная модель производственного процесса или системы, выполненная как комплекс символов, изображающих элементы системы (или процесс) с их взаимными связями.

Наглядно отображая структуру системы, мнемосхема облегчает оператору запоминание схем объектов, взаимосвязь между параметрами, назначение приборов и органов управления. В процессе управления мнемосхема является для оператора важнейшим источником информации о текущем состоянии системы, характере и структуре протекающих в ней процессов, в том числе связанных с нарушением технологических режимов, авариями и т.п.

Мнемосхемы эффективно используют в случаях, когда:

- управляемый объект имеет сложную технологическую схему и большое число контролируемых параметров;

- технологическая схема объекта может оперативно изменяться в процессе работы.

Мнемосхемы могут отражать как общую картину состояния системы, технологического процесса, так и состояние отдельных агрегатов, устройств, значения параметров и т.п. Мнемосхемы помогают оператору, работающему в условиях большого количества поступающей информации, облегчить процесс информационного поиска, подчинив его определенной логике, диктуемой реальными связями параметров контролируемого объекта. Они облегчают оператору логическую систематизацию и обработку поступающей информации, помогают осуществлению технической диагностики при отклонениях процесса от нормы, обеспечивают внешнюю опору для выработки оптимальных решений и формирования управляющих воздействий.

В основу построения мнемосхем положен ряд принципов, выработанных в процессе многолетней практики их применения. Один из основных - принцип лаконичности, согласно которому мнемосхема должна быть простой, не должна содержать лишних, затемняющих элементов, а отображаемая информация должна быть четкой, конкретной и краткой, удобной для восприятия и дальнейшей переработки.

Принцип обобщения и унификации предусматривает требование, согласно которому надо выделять и использовать наиболее существенные особенности управляемых объектов, т.е. на мнемосхеме не следует применять элементы, обозначающие несущественные конструктивные особенности системы, а символы сходных объектов и процессов необходимо по возможности объединять и унифицировать.

Согласно принципу акцента к элементам контроля и управления на мнемосхемах в первую очередь необходимо выделять размерами, формой или цветом элементы, наиболее существенные для оценки состояния, принятия решения и воздействия на управляемый объект.

Принцип автономности предусматривает необходимость обособления друг от друга участков мнемосхемы, соответствующих автономно контролируемым и управляемым объектам и агрегатам. Эти обособленные участки должны быть четко отграничены от других и согласно принципу структурности должны иметь завершенную, легко запоминающуюся и отличающуюся от других структуру. Структура должна отражать характер объекта и его основные свойства.

В соответствии с принципом пространственного соотнесения элементов контроля и управления расположение контрольно-измерительных и индикаторных приборов должно быть четко согласовано с расположением соответствующих им элементов управления, т.е. должен соблюдаться закон совместности стимула и реакции.

А принцип использования привычных ассоциаций и стереотипов предполагает применение на мнемосхемах таких условных обозначений параметров, которые ассоциируют с общепринятыми буквенными обозначениями этих параметров. Желательно применять, если это возможно, вместо абстрактных знаков символы, ассоциирующиеся с объектами и процессами.

Представление в виде мнемосхемы соответствует первому способу выделения систем, при котором из системы вычленяется множество составных элементов и между ними выявляются системообразующие межэлементные связи и отношения, придающие этому множеству целостность. Сложность построения мнемосхемы заключается в отсутствии жестких правил. В данном процессе нет ничего абсолютного и его нельзя свести к набору условных конструкций.

Рассмотрим основные положения к построению мнемосхемы, выработанные в процессе многолетней практики их применения.

1. Мнемосхема должна содержать только необходимые элементы для контроля и управления системой.
2. Отдельные элементы или группы элементов, наиболее существенные для контроля и управления, на мнемосхеме должны выделяться размерами, формой, цветом или другими способами.

3. Соединительные линии на мнемосхеме должны быть простой конфигурации, минимальной длины и иметь наименьшее число пересечений. Следует избегать большого числа параллельных соединительных линий, расположенных рядом.

4. Форма и размеры элементов мнемосхемы должны обеспечивать однозначное зрительное восприятие системы.

Рассмотрим основные принципы построения мнемосхемы.

1. Принцип простоты и адекватности. К мнемосхеме обычно предъявляются два противоречивых требования: простота и адекватность исследуемой системе. Степень сложности (или простоты) мнемосхемы определяется уровнем ее детализации, зависящим от принятых предположений, допущений, степени полноты и достоверности сведений об исследуемой системе: чем меньше предположений, допущений и достоверных сведений, тем выше уровень детализации и, следовательно, проще модель и, в то же время менее адекватна исследуемой системе. Процессы функционирования реальных систем практически невозможно описать полно и детально, что обусловлено существенной сложностью таких систем. Основная проблема при разработке мнемосхемы состоит в нахождении компромисса между простотой ее описания и необходимостью учета многочисленных особенностей, присущих реальной системе. Попытка построить единую универсальную мнемоническую модель сложной системы с большим количеством нагрузочных параметров, несомненно, обречена на неудачу ввиду ее необозримости и невозможности анализа. Для достижения компромисса между простотой модели, и в то же время ее адекватности исследуемой системе, на начальном этапе моделирования необходимо четко сформулировать цель моделирования, что позволит однозначно определить какие подсистемы, элементы и взаимосвязи между ними войдут в создаваемую модель, а какие будут носить затемняющий понимание характер.

2. Принцип выравнивания. Ни один элемент мнемосхемы не должен располагаться случайным образом, т.е. нельзя располагать элементы или

подсистемы где-либо на мнемосхеме только потому, что там присутствует свободное пространство.

3. Принцип уменьшения визуального шума. Визуальный шум является одним из основных препятствий для получения комплексного информационного представления о сложной системе.

Можно выделить два типа визуального шума:

1. Перегруженность. При работе с мнемосхемами, имеющими значительные размеры, большое количество всевозможных подсистем, элементов и бесчисленное множество соединительных линий различных цветов и яркостей, зрительная система подвергается большой нагрузке, что затрудняет восприятие и анализ системы в целом.

2. Фоновый шум. В качестве фона мнемосхемы рекомендуется применять малонасыщенные и нейтральные цвета средней частоты спектра. Не допускается использование в большом количестве цветов, которые затрудняют восприятие модели. Принцип использования привычных ассоциаций и стереотипов. Сущность данного принципа состоит в применении на мнемосхемах таких условных обозначений элементов, которые ассоциируются с общепринятыми буквенными обозначениями данных элементов. Желательно использовать, если это возможно, вместо абстрактных знаков и символов, графические образы, которые ассоциируются с элементами. Если изображение или его часть знакомо специалисту, то он сможет идентифицировать его даже по его фрагменту.

4. Принцип избыточности. Для улучшения восприятия структуры системы иногда бывает необходимо параллельное представление наиболее важных компонентов или взаимосвязей.

5. Принцип повтора. Некоторые компоненты стилового оформления должны повторяться, что делает мнемосхему более цельной, согласованной и упорядоченной. Повторяться могут маркеры списков, шрифтовое оформление (кегель, жирность), цветовые решения, пространственные взаимосвязи, толщина

Таблица стилей HTML Клиентский сценарий Серверный сценарий

соединительных линий, распределение свободного пространства и т.п. Положительная роль повтора очень сильна в одностраничных мнемосхемах, а в многостраничных без него не обойтись. Но необходимо избегать избыточного повтора, который будет подавлять смысловое содержание. [3]

1.2 Обзор SCADA-систем, используемых в производстве

На производстве существует достаточно большой выбор SCADA систем. Будут рассмотрены три SCADA-системы: RSView32, SIMATIC WinCC и Wonderware InTouch.

RSView32.

Мониторинг и управление машинами и процессами автоматизации требуют универсальности и масштабируемости для подключения к множеству открытых технологий. RSView@32™ представляет собой интегрированный компонентный HMI для мониторинга и управления автоматическими машинами и процессами. RSView32 расширяет представление с помощью открытых технологий, которые обеспечивают беспрецедентную возможность подключения к другим продуктам Rockwell Software, продуктам Microsoft и сторонним приложениям.

Основные функциональные возможности обеспечивают универсальность для удовлетворения целого ряда потребностей интеграции - от машины до производственной линии до всего предприятия. RSView32 - это открытая система, которая позволяет делиться данными заводов с другими производственными системами, обеспечивая информацию о производстве в режиме реального времени на предприятии.

Поддерживая интегрированную архитектуру Rockwell Automation Integrated Architecture и стратегию ViewAnyWare, RSView32 является частью масштабируемого и единого комплекса программно-аппаратных средств

мониторинга и управления, разработанных для расширения автономных приложений машинного уровня посредством приложений человеко-машинного интерфейса (HMI) контролирующего уровня по всей сети. Этот комплекс предлагает вам общую среду разработки, многократное использование приложений и архитектуру, таким образом, вы можете повысить производительность, снизить эксплуатационные расходы и улучшить качество.

RSView32 является первым программным обеспечением человеко-машинного интерфейса, предназначенного для того, чтобы:

- открывать графические дисплеи как OLE-контейнеры для элементов управления ActiveX® — с тысячами сторонних элементов управления ActiveX на ваш выбор, которые вы можете вставить прямо в ваш проект в качестве уже готовых решений;

- разрабатывать объектную модель для раскрытия части ее основных функциональных возможностей, позволяя RSView32 легко взаимодействовать с другими программными продуктами, основанными на использовании компонентов;

- интегрировать популярный Microsoft Visual Basic® for Applications (VBA) в качестве встроенного языка программирования, позволяя практически неограниченные возможности по настройке ваших проектов RSView32;

- поддерживать стандарты OPC как для сервера, так и для клиента, для быстрой и безопасной связи с широким спектром аппаратных средств;

- реализовать технологию добавляемой архитектуры (AOA) для расширения функциональности RSView32 и интеграции новых возможностей прямо в ядро RSView32.

Рисунок 1.1 показывает пример проекта, который может быть создан в системе RSView32.

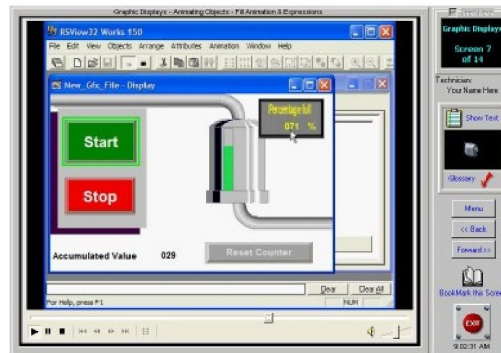


Рисунок 1.1 – Проект, созданный в системе RSView32

SIMATIC WinCC.

SIMATIC WinCC – мощная среда разработки верхнего уровня управления АСУ ТП с централизованным контролем и сбором данных, система SIMATIC WinCC (Windows Control Center) - это компьютерная система человеко-машинного интерфейса, работающая под управлением операционных систем Windows и предоставляющая широкие функциональные возможности для построения систем управления различного назначения и уровней автоматизации.

Система SIMATIC WinCC разработана для решения задач визуализации и оперативного управления в различных областях промышленного производства. Система оснащена мощным интерфейсом для связи с процессом, пригодна для работы со всем спектром изделий SIMATIC, обеспечивает парольный доступ к управлению процессом, обладает высокой производительностью. Базовая конфигурация системы обладает высокой универсальностью и может быть использована для построения систем управления самого разнообразного назначения.

Программа оснащена мощным интерфейсом для связи с процессом, пригодна для работы со всем спектром систем автоматизации SIMATIC, обеспечивает защиту данных и возможность их архивирования, обладает высокой производительностью, поддерживает резервированные структуры управления. SCADA система является информационным центром, обеспечивающим

поддержку принципа вертикальной интеграции в масштабах всей компании. Базовая конфигурация системы обладает высокой универсальностью и может быть использована для построения систем управления самого разнообразного назначения. Специализированные решения для конкретных областей промышленного производства могут разрабатываться на основе дополнительного программного обеспечения.

WinCC может работать на любом компьютере, отвечающими системным требованиям данного пакета. В сочетании с промышленными компьютерами SIMATIC Panel IPC и/или SIMATIC Rack IPC появляется возможность построения высокопроизводительных систем, обеспечивающих возможность непрерывного 24-часового функционирования в промышленных условиях или в офисе.

Преимуществом системы является наличие всех функций, присущих SCADA системам – для полной графической визуализации процесса и его состояний, для создания отчетов и квитирования событий, для регистрации значений измеряемых величин и сообщений системы, для регистрации и архивирования данных, для управления пользователями и их правами доступа. Система непрерывно регистрирует последовательность операций и событий, влияющих на качество, что позволяет осуществлять постоянный контроль качества.

Усовершенствованные функции проектирования заметно сокращают время и усилия, необходимые для разработки и обучения работе с системой: удобный для пользователя, объектно-ориентированный графический редактор (с индивидуальными настройками и возможными расширениями, создаваемыми с помощью Visual Basic для приложений (VBA)), обширные библиотеки, модульная технология, возможность оперативного изменения с использованием проектирования в режиме online, инструментальные средства проектирования для обработки больших массивов данных, прозрачность системы, благодаря списку перекрестных ссылок.

Основные возможности:

					EEEE 000000 000 000 ПП	Лист
Изм.	Лист	№ докум.	Подпис	Дата		4

1. Визуализация техпроцесса (Graphic Designer).
2. Конфигурирование и настройка связи с контроллерами различных производителей (Tag Management).
3. Отображение, архивирование и протоколирование сообщений от технологического процесса (Alarm Logging).
4. Отображение, архивирование и протоколирование переменных (Tag Logging).
5. Расширение возможностей системы за счет использования скриптов на языках ANSI C, VBS и VBA.
6. Проектирование системы отчетности (Report Designer).[5]

На рисунке 1.2 показан пример проекта, который создан в системе SIMATIC WinCC.

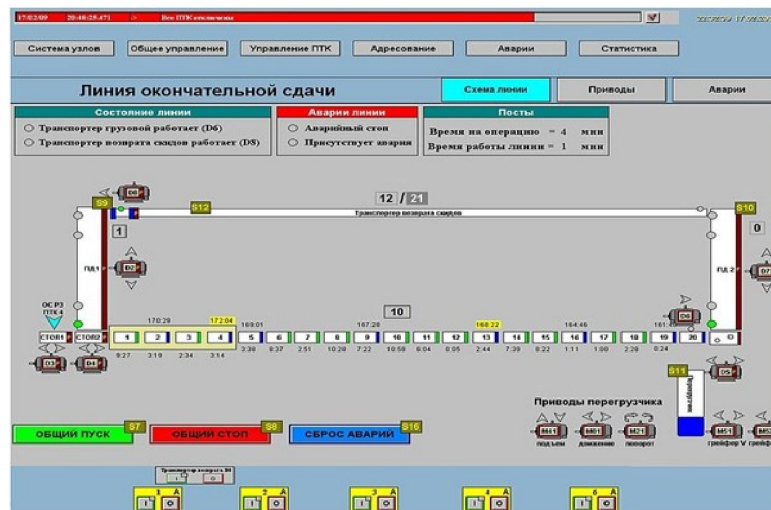


Рисунок 1.2 – Проект, созданный в системе SIMATIC WinCC

Wonderware InTouch.

Wonderware InTouch HMI – это открытый и расширяемый человекомашинный интерфейс с передовыми графическими возможностями,

мощными средствами разработки и гибкой архитектурой для быстрого создания и развертывания приложений автоматизации, позволяющий пользователям связываться с практически любыми промышленными устройствами и системами.

InTouch HMI позволяет перевести на качественно новый уровень оптимизацию, контроль и управление производственными процессами.

На протяжении более 25 лет, Wonderware InTouch был номером один в мире человеко-машинных интерфейсов (HMI). InTouch предлагает:

- легендарную простоту использования, лидирующую на рынке инноваций;
- непревзойденную защиту инвестиций;
- блестящую графику;
- непревзойденную связность;
- лучшую поддержку в отрасли и широчайшую партнерскую экосистему.

SCADA система InTouch – это один из самых популярных в мире программных пакетов визуализации человеко-машинного интерфейса(HMI). Wonderware intouch, а точнее первые его разработки появились около 20 лет назад, и за это время успели пройти испытание на множестве реальных промышленных объектов. Попробуем разобраться, чем отличается система intouch от прочих SCADA конфигураторов и чем она примечательна.

SCADA система Wonderware intouch поддерживает интеграцию со всеми крупными производителями рынка систем и решений автоматике включая Siemens ABB, Rockwell Automation и многими другими. Приложения, созданные в этой среде успешно используются в самых различных отраслях: от добычи нефти до производства продуктов питания. Библиотека объектов с индустриальной графикой поистине огромна и позволит создать мощные и информативные экраны человеко-машинного интерфейса (HMI). Способствует этому и большой набор программных сценариев.

Wonderware intouch состоит из двух частей: среды разработки и среды исполнения. В среде разработки WindowMaker рисуют мнемосхемы и описывают

сценарии, после чего загружаются в среду исполнения. Серверы ввода/вывода данных ориентирован на конкретное оборудование. Используются как правило стандартные протоколы обмена данными: DDE, OPC, TCP/IP и тд. Заявлено, что SCADA система Wonderware intouch способна работать с более чем 120000 сигналов и параметров. Intouch содержит в себе встроенные исторические и временные тренды, элементы управления Microsoft ActiveX и .NET, поддерживает Microsoft Remote Desktop Services. За счет работы как сервиса сетевой ОС Windows NT функционирование программы не нарушается процессом регистрации пользователей.[6]

На рисунке 1.3 изображен проект, который создан в системе Wonderware InTouch.

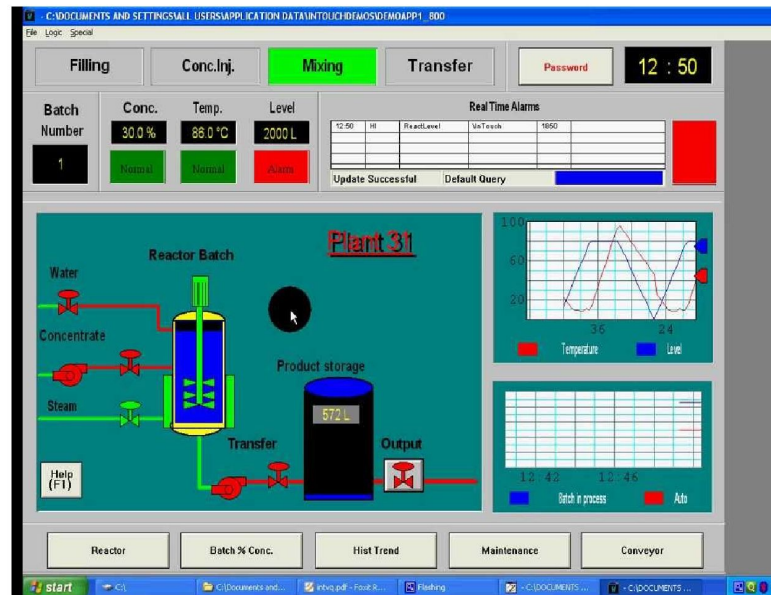


Рисунок 1.3 – Проект, созданный в системе Wonderware InTouch

1.3 Выбор SCADA-системы для разработки мнемосхемы

SCADA-система, как и любое программное обеспечение, выдвигает некоторые требования к аппаратному и программному обеспечению. Это определяет конфигурацию программно-технических средств, на которых будет работать разработанное прикладное программное обеспечение.

Во-первых, это требование к операционной системе. Большинство SCADA-систем работают на MS Windows, но есть и системы, работающие на Linux, Mac OS и других.

Во-вторых, требования к аппаратному обеспечению: требования к процессору, объему памяти, сетевым возможностям для связи с другими уровнями АСУ ТП. Сюда же относится возможность применения «ключа» платной версии SCADA (программный ключ, аппаратный USB-ключ или другой).

К техническим требованиям также относятся требования ко встроенным командным языкам, поддерживаемым системам управления баз данных и др.

Распространенность.

Один из критериев для выбора SCADA-системы является ее распространенность. Всегда проще и дешевле найти специалистов, работающих с распространенными SCADA-системами, чем с какими-нибудь экзотическими и малоизвестными.

Стоимость.

Один из основных критериев выбора SCADA-системы. Стоимость зависит от маркетинговой стратегии производителей систем. Бывают бесплатные, условно-бесплатные и платные системы.

Бесплатные и условно-бесплатные системы, как правило, имеют ограничения либо по количеству тегов (наборов данных, передаваемых от нижнего уровня SCADA-системе и обратно), либо по времени непрерывной работы. Обычно это до 50 тегов или до 1 часа работы.

В платных системах плата обычно берется за фиксированное количество тегов (например, 100, 500, 5000 и т.д.), а также за дополнительные модули, которые расширяют функционал SCADA-системы. Причем среда разработки, как правило, бесплатная.

Функциональные возможности.

Как уже говорилось, любая SCADA-система располагает функционалом, позволяющем разрабатывать программное обеспечение верхнего уровня любой АСУ ТП. Различия могут быть в присутствии в той или иной SCADA-системе специфических функций. На это надо обращать внимание.

Открытость.

SCADA-система называется открытой, если для нее описаны все форматы данных и процедуры обмена данными. Это позволяет подключать к такой системе дополнительные, специально разработанные компоненты. Такими компонентами могут быть программные модули, процедуры, которые выполняют специфические функции. Также это могут быть драйвера нестандартных устройств, отсутствующие в библиотеке SCADA-системы. Если в таких компонентах есть необходимость, то надо выбирать открытую SCADA-систему.

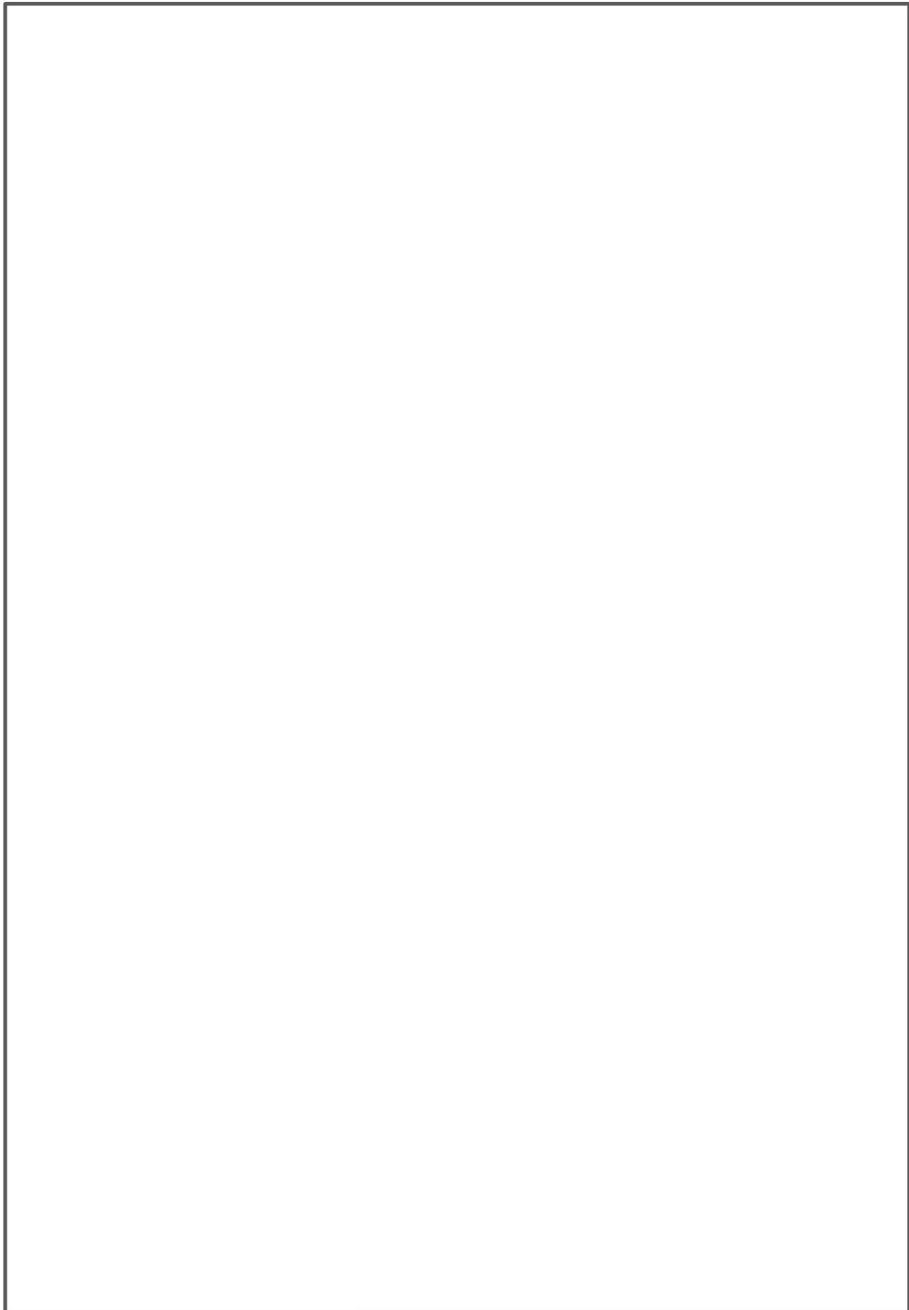
Эксплуатационные характеристики.

Этот критерий определяет удобство пользования той или иной SCADA-системой при разработке прикладного программного обеспечения. Сюда входят:

- удобный интерфейс;
- локализованная и полная техническая документация;
- качественная поддержка системы со стороны производителя[7].

Популярность в родном городе. Самый главный критерий – это то, что почти все объекты на Енакиевском металлургического завода работают под управлением этой SCADA-системы. То есть, если я выбрал бы другую систему, то потом для работы на заводе пришлось бы переучиваться на SIMATIC WinCC.

Исходя из всех выше критериев, для разработки SCADA-системы была выбрана SCADA-система SIMATIC WinCC.



					ЕЕЕЕ 000000 000 000 ПП	Лист
Изм.	Лист	№ докум.	Подпис	Дата		4

РАЗДЕЛ 2. РАЗРАБОТКА МНЕМОСХЕМЫ ВИЗУАЛИЗАЦИИ ТЕХНОЛОГИЧЕСКОГО ОБЪЕКТА С ПОМОЩЬЮ SCADA-СИСТЕМЫ

2.1 Постановка задачи создания мнемосхемы технологического объекта

Система водоснабжения включает комплекс сооружений и устройств для забора воды из источника водоснабжения, её очистки, хранения, подачи и распределения между потребителями. На металлургических заводах применяют прямоточное, последовательное и оборотное водоснабжение. Поэтому имеются водоводы свежей технической воды, оборотной или повторно используемой воды, производственного (технического) водоснабжения, снабжения питьевой водой и пожарного водоснабжения (объединенного с производственным или питьевым). При прямоточном водоснабжении отработанная вода после очистки и охлаждения сбрасывается в водоём, из которого была взята. Последовательное водоснабжение предусматривает повторное использование воды рядом потребителей с учетом их требований к качеству воды с последующим сбросом. В оборотных циклах водоснабжения вода подвергается очистке от взвешенных частиц и охлаждению, и снова используется. Свежая вода потребляется только для компенсации потерь, составляющих порядка 5% от количества воды, находящейся в обороте. Очистка воды производится в отстойниках или гидроциклонах, а также аппаратах тонкой очистки (фильтрах), а охлаждение воды – в прудах-охладителях, брызгальных бассейнах, в башенных или вентиляторных градирнях.

Вентиляторные градирни применяют при необходимости устойчивого и глубокого охлаждения воды. Отличаются вентиляторные градирни от башенных отсутствием вытяжных башен. Движение охлаждающего воздуха навстречу охлаждаемой воде организуется с помощью вентиляторов. В градирнях этого типа

разница температур охлаждаемой и охлажденной воды составляет 10-15°C. В вентиляторных градирнях осуществляется автоматическое поддержание температуры охлажденной воды на заданном уровне путём изменения производительности вентиляторов.

Механический фильтр.

Практически ни одна система водоподготовки не исключает применения в своем составе блока механической фильтрации. Качество механической обработки потока воды, в конечном счете, позволяет снизить нагрузку на последующие элементы системы фильтрации.

Основными характеристиками фильтров механической очистки (ФМО) воды являются степень задержания механических примесей по фракционному составу и пропускная способность при минимальном гидравлическом сопротивлении (производительность). Кроме того, имеют место такие характеристики, как рабочая температура воды и рабочее давление воды. Производительность фильтров ограничивается, как правило, линейными зависимостями подачи воды от потерь давления на элементе фильтрации при определенном давлении входной воды. Рекомендованные производителями фильтров механической очистки воды рабочие характеристики применения фильтров основаны на опытных данных, полученных при испытаниях оборудования, исключающих возможность срыва потока воды от ламинарного к турбулентному и обеспечения регламентированной скорости потока через определенный тип фильтрующей загрузки технологический процесс осветления воды фильтрованием реализуется главным образом методом адгезионного объемного фильтрования в насыпных вертикальных осветительных фильтрах.

Фильтр состоит из цилиндрического корпуса с приваренными к нему сферическими днищами. Внутри фильтра расположены слой фильтрующего материала и дренажно-распределительные устройства, предназначенные для равномерного распределения и сбора воды по площади поперечного сечения фильтра. Верхнее дренажное устройство выполнено в виде отбойного щита,

гасящего энергию потока поступающей воды, а нижнее состоит из коллектора с боковыми отводами, снабженными для отвода воды и в качестве препятствия для выноса фильтрующего материала специальными колпачками или щелевыми отверстиями шириной 0,4 мм.

Фильтрующий материал насыпных фильтров должен обладать надлежащим гранулометрическим составом, достаточной механической прочностью и химической стойкостью зерен. Таким требованиям удовлетворяет катионит сульфуголь. Размеры зерен угля должны составлять 0,6 - 1,4 мм для однослойного фильтрования. В соответствии с требованиями механической прочности годовой износ фильтрующего материала не должен превышать 2,5 %. Высота фильтрующего материала в осветительных фильтрах составляет около 1 м.

Работа осветительных фильтров подразделяется на три периода:

- полезная работа фильтра по осветлению воды;
- взрыхляющая промывка фильтрующего материала;
- спуск первого фильтрата.

Полезная работа насыпного фильтра реализуется при скоростях фильтрования воды до 10 м/ч при предварительной ее обработке в осветлителях и 4 - 5 м/ч без предварительной обработки.

Во время работы осветительных фильтров необходимо поддерживать постоянную скорость фильтрования, контролировать перепад давления на слое фильтрующего материала и расход воды, отбирать пробы исходной воды и фильтрата для определения прозрачности. [8]

Установка состоит из множества труб, насосов. Имеет два фильтра и пять насосов. Схема приведена на рисунке 2.1

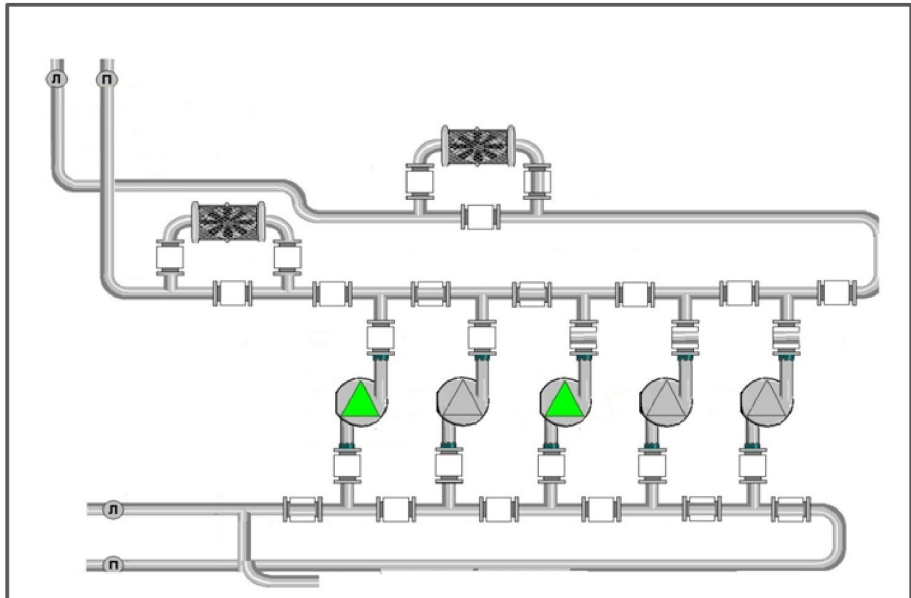


Рисунок 2.1 – Схема повысительной насосной станции с фильтровальной установкой

2.2 Отображение состояний механизмов и датчиков на технологических мнемосхемах

Для разработки мнемосхемы служит модуль Graphic Designer(графический дизайнер). С его помощью можно разрабатывать мнемосхемы с поддержкой централизованно изменяемых шаблонов дизайна, цветовой палитры объектов пользователя.

В графическом дизайнера присутствует множество инструментов. От простого изменения заднего фона кадра, до сложных механизмов. На рисунке 2.2 показан интерфейс графического дизайнера.

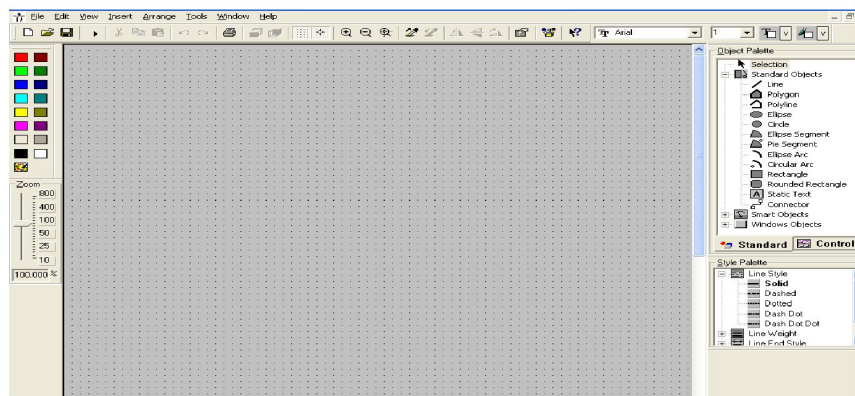


Рисунок 2.2 – Окно редактора кадра

Графический дизайнер содержит следующие компоненты: палитры, предназначенные для создания и редактирования графических объектов, палитра цвета, палитра объектов, палитра стилей, палитра выравнивания, палитра масштабирования, палитра шрифтов. Также он содержит палитры и панели для управления графическим дизайнером: панель меню, стандартная палитра, строка состояния, панель слоев и диалоговое окно для установки и изменения свойств объекта: Object Properties (Свойства объекта).

На палитре цвета, используя мышью, для выделенного объекта можно выбрать один из 16 стандартных цветов, основной цвет, или самостоятельно настраиваемый цвет. Например, щелчок кнопкой мыши на панели цвета изменяет один из следующих цветов: цвет заливки для контурных объектов (например, прямоугольников), цвет линий для объектов, состоящих из линий (например, ломаных кривых), цвет фона текстовых объектов. На рисунке 2.3 изображена палитра цвета.



Рисунок 2.3 – Палитра цветов

Палитра объектов. Используя Палитру объектов, можно выбрать объект, который будет создан в первом видимом слое рабочей области Графического дизайнера. Для этих целей в Графическом дизайнере предназначен слой 0. Объекты сгруппированы в соответствии с областью их применения: стандартные объекты (Standard objects), “интеллектуальные” объекты (Smart Objects) и объекты Windows(рисунок 2.4).

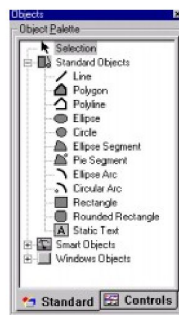


Рисунок 2.4 – Палитра объектов

Палитра стиля. Используя палитру стиля, можно изменять внешний вид выбранного объекта. В зависимости от типа объекта предлагаются различные типы стилей. Например: различные стили линий/границ, выбор ширины линий/границ, стили окончания линий для объектов, состоящих из линий, различные стили заливки. На рисунке 2.5 показана палитра стиля.



Рисунок 2.5 – палитра стилей






Палитра выравнивания. С помощью Палитры выравнивания можно произвести одно из следующих действий: изменение абсолютного положения одного или нескольких объектов, изменение положения выбранных объектов относительно друг друга, приведение высоты и ширины нескольких объектов к одному значению. В распоряжении пользователя имеются следующие функции выравнивания, которые приведены в таблице 1.

Таблица 1. Функции выравнивания

Пиктограмм	Описание
а	2
	Выравнивание объектов по левой стороне (Меню Arrange - Align - Left – Распределить – Выравнивание - Слева)
	Выравнивание объектов по правой стороне (Меню Arrange - Align – Right – Распределить – Выравнивание - Справа)
	Выравнивание объектов по верхней границе (Меню Arrange - Align – Top – Распределить – Выравнивание – Сверху)
	Выравнивание объектов по нижней границе (Меню Arrange – Align – Bottom – Распределить – Выравнивание – Снизу)
	Горизонтальная центровка объектов (Меню Arrange – Align – Center Horizontally - Распределить – Выравнивание – Горизонтальная центровка)
	Вертикальная центровка объекта (Меню Arrange – Align – Center Vertically – Распределить – Выравнивание – Вертикальная центровка)

Продолжение таблицы 1.

1	2
---	---

	Распределить объекты равномерно по горизонтали (Меню Arrange – Align – Space Horizontally – Распределить – Выравнивание – Распределить по горизонтали)
	Распределить объекты равномерно по вертикали (Меню Arrange – Align – Space Vertically – Распределить – Выравнивание – Распределить по вертикали)
	Привести ширину объектов к одному значению (Меню Arrange – Align – Same Width – Распределить – Выравнивание – Привести ширину)
	Привести высоту объектов к одному значению (Меню Arrange – Align – Same Height – Распределить – Выравнивание – Привести высоту)
	Привести ширину и высоту объектов к одному значению (Меню Arrange – Align – Same Width and Height)

Палитра масштабирования. Палитра масштабирования позволяет установить коэффициент масштабирования активного окна с помощью ползунка или кнопки. Текущий коэффициент масштабирования отображается под ползунком. Можно также производить дискретное изменение масштаба с помощью стандартных функций панели инструментов. На рисунке 2.6 показана палитра масштабирования.





Рисунок 2.6 – Палитра масштабирования










Стандартная панель инструментов Стандартная панель инструментов содержит кнопки, с помощью которых, щелкнув на них мышью, можно выполнять команды, используемые наиболее часто. На панели инструментов имеются следующие кнопки, которые обозначены в таблице 2.

Таблица 2. Описание кнопок на панели инструментов.

Кнопка	Описание
1	2
	New File - Создать файл (Меню File - New - Файл - Создать)
	Open File - Открыть файл (Меню File - Open ... - Файл - Открыть...)
	Save File - Сохранить файл (Меню File - Save - Файл - Сохранить)
	Runtime - Запуск режима Run-Time (Меню File - Runtime - Файл - Запустить Run-Time)
	Cut - Вырезать (Меню Edit - Cut - Правка - Вырезать)
	Copy - Копировать (Меню Edit - Copy - Правка - Копировать)

Продолжение таблицы 2.

1	2
	Paste - Вставить (Меню Edit - Paste - Правка - Вставить)
	Undo - Отменить (Меню Edit - Undo - Правка - Отменить)
	Redo - Повторить

	(Меню Edit - Redo - Правка - Повторить)
	Print File - Печатать файл (Меню File - Print ... - Файл - Печатать...)
	Place on Top within the Layer - Разместить сверху в пределах слоя (Меню Arrange - Within the Layer - Place on Top - Распределить - В пределах слоя - Разместить сверху)
	Place on Bottom within the Layer - Разместить снизу в пределах слоя (Меню Arrange - Within the Layer - Place on Bottom - Распределить - В пределах слоя - Разместить снизу)
	Grid On/Off - Сетка включена/выключена (Меню View - Grid ... - Вид - Сетка...)
	Snap to Grid - Привязать к сетке (Меню View - Grid ... - Вид - Сетка...)
	Enlarge View - Увеличить изображение (Меню View - Zoom - Zoom In - Вид - Масштабировать - Увеличить масштаб)
	Reduce View - Уменьшить изображение (Меню View - Zoom - Zoom Out - Вид - Масштабировать - Уменьшить масштаб)
	Enlarge Segment - Увеличить сегмент (Меню View - Zoom - Area - Вид - Масштабировать - Область)

Палитра шрифтов. С помощью палитры шрифтов можно произвести быстрое изменение цвета шрифта, самого шрифта, а также цвета линий стандартных объектов. Таблица 3 иллюстрирует палитру шрифтов.

Таблица 3. Палитра шрифтов.

Пиктограмма	Описание
	Изменение шрифта
	Изменение размера шрифта
	Изменение цвета шрифта
	Изменение цвета линии

Панель слоев. С помощью панели слоев можно выбрать, какой из 16 слоев (0...15) будет видимым, щелкнув на нем. Объекты создаются в первом видимом слое. В Графическом дизайнера для этих целей по умолчанию используется слой 0.[9]

Панель тегов. Панель тегов позволяет простым образом подключать теги к атрибутам объекта, чтобы сделать их динамичными. Теги выбираются на панели тегов. Подключение тега к атрибуту выполняется путём перетаскивания и размещения выбранного тега на атрибуте с помощью мышки.

Для создания мнемосхемы, потребовалось множество объектов. Эти объекты можно найти в библиотеке – View -> Library -> Global Library. На рисунке 2.7 показана библиотека объектов.

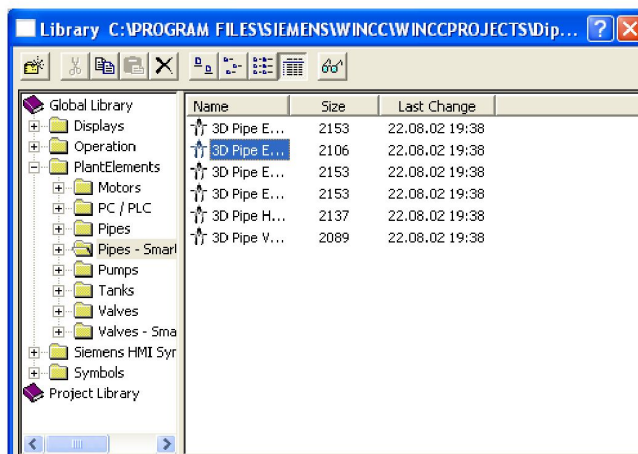


Рисунок 2.7 – Библиотека объектов

Используя объекты из графического дизайнера, создаём мнемосхему технологического объекта. Все используемые объекты приведены в таблице 4.

Таблица 4. Перечень объектов и их состояний.

Отображение на кадре	Обозначение
	Насос не включен.
	Насос включен.
	Задвижка закрыта.
	Задвижка открыта.
	Труба.
	Фильтр.

2.3 Технологические параметры мнемосхемы

В WinCC Explorer (Проводнике WinCC) теги создаются в Tag Management (Менеджере тегов). Существует различие между не связанными с драйвером процесса тегами, называемыми Internal Tags (Внутренними тегами), и тегами, связанными с драйвером процесса, называемыми WinCC Tags (Тегами WinCC), или External Tags (Внешними тегами). Для внутренних тегов нет ограничений на их количество. В то же время максимально возможное количество тегов WinCC зависит от приобретенной лицензии.

При обработке большого количества данных и, следовательно, большого количества тегов, лучше объединять эти теги в группы. Только таким образом можно отслеживать правильность конфигурации в крупномасштабных проектах. Тем не менее, группы тегов никак не гарантируют уникальность используемых тегов. Это делается исключительно посредством имен тегов.

Теги, используемые в WinCC, представляют собой либо реальные значения – такие как, например, уровень заполнения бака с водой - или внутренние значения, которые вычисляются или моделируются внутри WinCC.

Теги процесса (external tags) служат для обмена данными между WinCC и контроллерами. Каждый тег процесса в WinCC соответствует определенному значению процесса в памяти одного из контроллеров. В режиме исполнения WinCC считывает область данных, в которой хранится это значение процесса, из контроллера и, таким образом, определяется значение тега процесса. Например, уровень заполнения бака с водой определяется датчиком уровня заполнения, и это значение записывается в память ПЛС. С помощью установленного соединения – канала связи – значение уровня заполнения передается в WinCC.

Теги, не получающие значения от процесса, создаются в папке internal tags.

В данном дипломном проекте используются только внутренние теги. По сравнению с тегами процесса, у внутренних тегов относительно простая структура.

Создаются группы тегов так: на главном окне WinCC (WinCC Explorer) подводим курсор к знаку + возле пункта Tag Management и кликаем левой

кнопкой мыши. Откроется подпункт Internal Tags. В нём уже присутствуют некоторые системные теги (см. рисунок 2.8).

Name	Type	Parameters	Last Change
Script	Tag group		5/19/2018 2:40:31 PM
TagLoggingRt	Tag group		5/19/2018 2:40:31 PM
@CurrentUser	Text tag 8-bit ...		5/19/2018 2:40:31 PM
@DeltaLoaded	Unsigned 32-b...		5/19/2018 2:40:31 PM
@LocalMach...	Text tag 8-bit ...		5/19/2018 2:40:31 PM
@Connecte...	Unsigned 16-b...		5/19/2018 2:40:31 PM
@Redundan...	Unsigned 16-b...		5/19/2018 2:40:31 PM
@Datasourc...	Text tag 16-bi...		5/19/2018 2:40:31 PM
@ServerName	Text tag 16-bi...		5/19/2018 2:40:31 PM
@CurrentUs...	Text tag 16-bi...		5/19/2018 2:40:31 PM

Рисунок 2.8 – Внутренние теги проекта

Далее, кликаем правой кнопкой мыши на подпункте Internal tags и выбираем New Group. Откроется окно свойств группы тегов (рисунок 2.9).

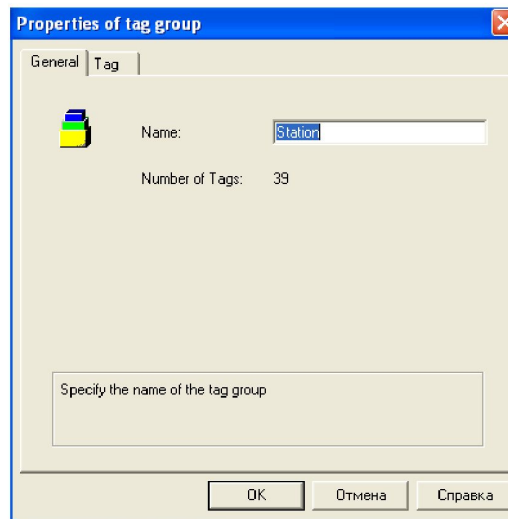


Рисунок 2.9 – Окно свойств группы тегов

Вводим имя Station для группы тегов и нажимаем кнопку ОК. Группа тегов создавалась и теперь в эту группу можно будет помещать необходимые теги проекта.

Для функционирования объектов на мнемосхеме, нужны теги. Они содержат значения для объектов на мнемосхеме. Например, если значение 1 – задвижка открыта, 0 – закрыта.

Чтобы создать тег, нажимаем правую кнопку мыши на уже созданной группе тегов (Station) и выбираем пункт New Tag. Открылось окно свойств тега (рисунок 2.10).

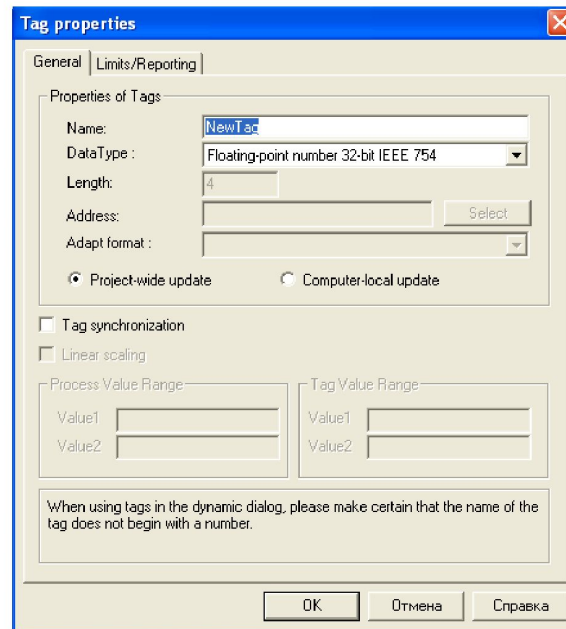


Рисунок 2.10 – Окно свойств тега

Окно свойств тега содержит в себе две вкладки. Во вкладке General предлагается ввести имя тега, тип значений для него и так далее. Вкладка Limits/Reporting предназначена для ввода стартовых, минимальных и

максимальных значений тега, если тег имеет тип значений числовой. На рисунке 2.11 изображена вкладка Limits/Reporting.

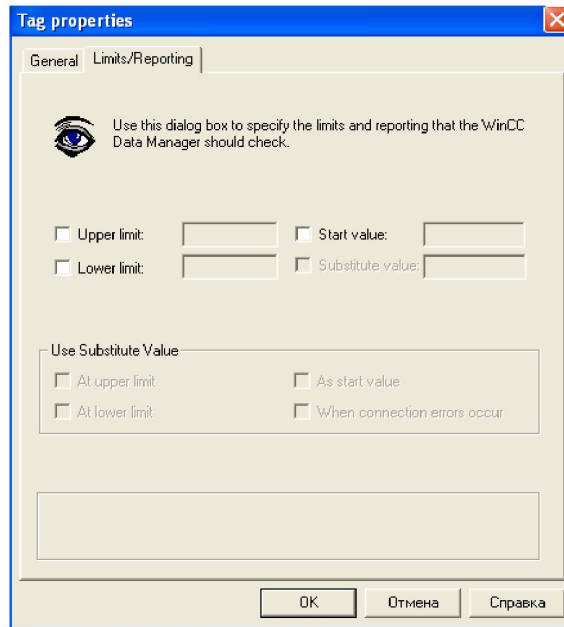


Рисунок 2.11 – вкладка Limits/Reporting окна Tag properties

Задаём имя тегу Engine 1 и значения, и нажимаем на кнопку ОК. Тег создан. Таким образом, были созданы теги для всех необходимых объектов на мнемосхеме. В таблице 5 приведены все теги, которые используются в проекте.

Таблица 5. Список тегов

Наименование тега	Тип значения тега	Привязанность к объекту
Engine 1	Boolean	Насос(951)
Engine2	Boolean	Насос(950)
Engine3	Boolean	Насос(949)
Engine4	Boolean	Насос(948)

Продолжение таблицы 5.

1	2	3
Engine5	Boolean	Насос(947)
F 1	Float	Объём воды в трубе 1
F 2	Float	Объём воды в трубе 2
P 1	Float	Давление воды
P 2	Float	Давление воды
P 3	Float	Давление воды
P 4	Float	Давление воды
Zadv	Boolean	Задвижка(977)
Zadv 1	Boolean	Задвижка(976)
Zadv 2	Boolean	Задвижка(975)
Zadv 3	Boolean	Задвижка(973)
Zadv 4	Boolean	Задвижка(972)
Zadv 5	Boolean	Задвижка(971)
Zadv 6	Boolean	Задвижка(970)
Zadv 7	Boolean	Задвижка(969)
Zadv 8	Boolean	Задвижка(968)
Zadv 9	Boolean	Задвижка(961)
Zadv 10	Boolean	Задвижка(960)
Zadv 11	Boolean	Задвижка(959)
Zadv 12	Boolean	Задвижка(958)
Zadv 13	Boolean	Задвижка(957)
Zadv 14	Boolean	Задвижка(956)
Zadv 15	Boolean	Задвижка(955)
Zadv 16	Boolean	Задвижка(954)
Zadv 17	Boolean	Задвижка(953)
Zadv 18	Boolean	Задвижка(952)
Zadv 19	Boolean	Задвижка(967)

Продолжение таблицы 5.

1	2	3
Zadv 20	Boolean	Задвижка(966)
Zadv 21	Boolean	Задвижка(965)
Zadv 22	Boolean	Задвижка(964)
Zadv 23	Boolean	Задвижка(963)
Zadv 24	Boolean	Задвижка(962)
Zadv 25	Boolean	Задвижка(980)
Zadv 26	Boolean	Задвижка(979)
Zadv 27	Boolean	Задвижка(978)

2.4 Разработка мнемосхемы технологического объекта

При запуске среды разработки (WinCC), программа предлагает выбрать один из трёх вариантов создания проекта:

- Single-User Project (проект для одного пользователя);
- Multi-User Project (проект для двух и более пользователей);
- Client Project (клиентовский проект).

А четвёртый вариант (Open an Existing Project) открывает ранее созданный проект).

Варианты создания проекта изображены на рисунке 2.12.

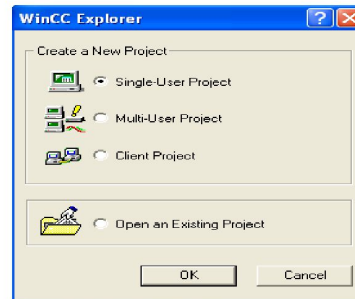


Рисунок 2.12 – Варианты создания проекта

Выбираем первый вариант (Single-User Project). После выбора пункта, вводим имя проекту. Рисунок 2.13 иллюстрирует окно ввода названия проекта и его создания.

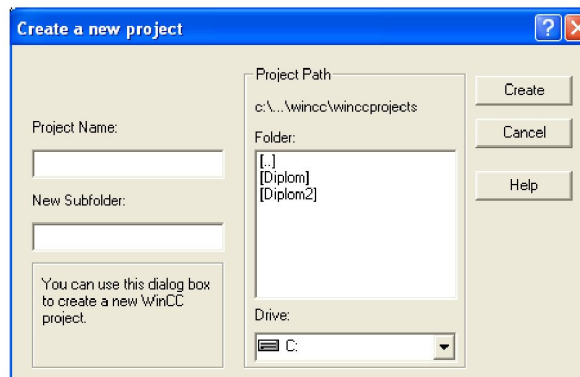


Рисунок 2.13 – Окно ввода названия проекта и его создания

Затем, когда название заполнено, нажимаем на кнопку Create для создания проекта. Ждём некоторое время, пока проект создастся. После его создания, появляется окно с компонентами и настройками проекта (см. рисунок 2.14).

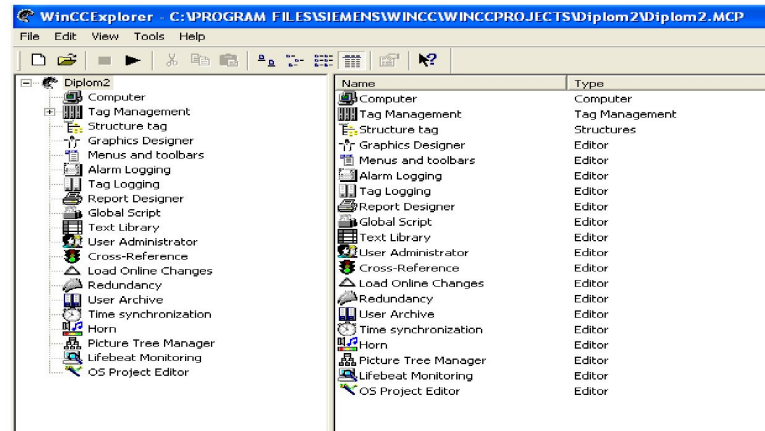


Рисунок 2.14 – Окно с компонентами и настройками проекта

После создания проекта, создаём кадр процесса. Для этого, подводим курсор к пункту Graphic Designer и нажимаем на правую кнопку мыши. Высветилось меню (рисунок 2.15), в котором выбираем пункт New Picture. Кадр создан.

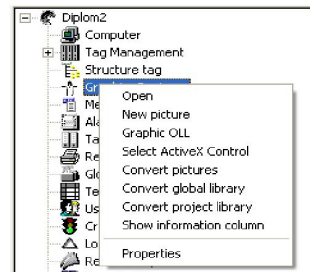


Рисунок 2.15 – Пункты меню при создании нового кадра процесса

Переименовываем кадр в Start.PDL. Нажимаем правой кнопки мыши на сам кадр, выбираем пункт Rename Picture и переименовываем кадр в Start.

Открываем только что созданный кадр. Он пустой. Создаём мнемосхему с помощью объектов, которые размещены в библиотеке объектов. На рисунке 2.16 изображена уже готовая мнемосхема технологического объекта “Повысительная насосная станция с фильтровальной установкой”.

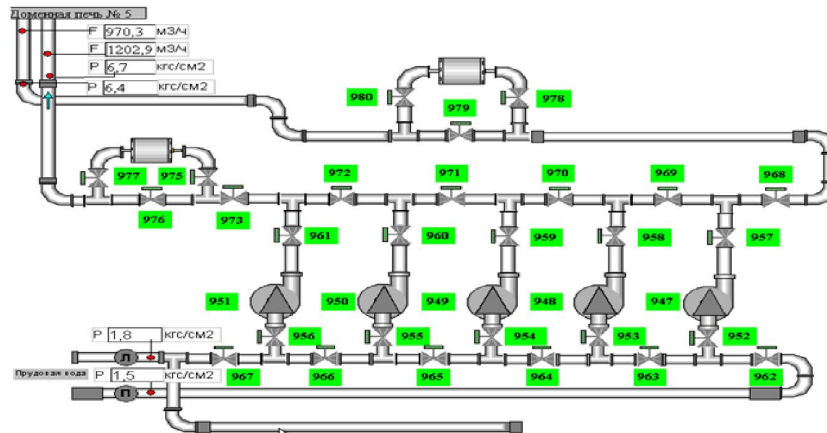


Рисунок 2.16 – Мнемосхема технологического объекта “Повысительная насосная станция с фильтровальной установкой” без включенных объектов

Чтобы объекты на мнемосхеме изменяли цвет в зависимости от того, активен ли объект или нет, необходимо привязать к объектам теги.

Тег можно размещать в двух местах: над объектом в кадре и над атрибутом во вкладке "Properties" ("Свойства") диалогового окна "Object Properties" ("Свойства объекта"). На рисунке 2.17 показаны свойства объекта.

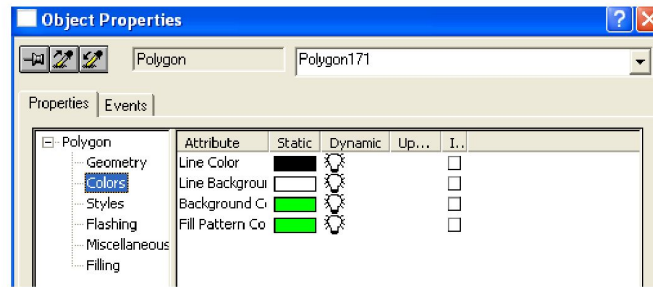


Рисунок 2.17 – Окно свойств объекта

Для привязки тега к объекту, выбираем пункт Colors, далее пункт Background Color и нажимаем правую кнопку мыши на лампочке. Из предложенных вариантов, выбираем Dynamic Dialog. Окно Dynamic Dialog показано на рисунке 2.18.

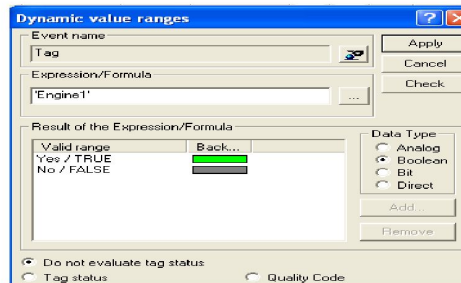


Рисунок 2.18 – окно Dynamic Dialog

Выбираем нужный тег, нажимая на кнопку выбора тега. Открылось окно изменения события (см. рисунок 2.19).

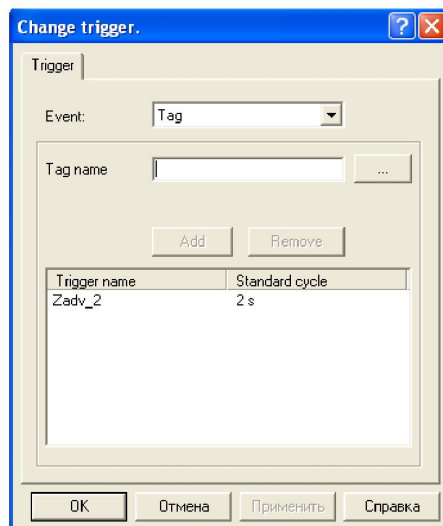


Рисунок 2.19 – окно изменения события.

Из предложенного списка(Events) выбираем Tag, далее вводим имя тега (Zadv_2). Нажимаем кнопку ОК. В меню Expression/Formula, снова выбираем тот же тег. В Data Type выбираем тип Boolean и цвета для каждого из значений (зелёный – открыта задвижка, серый - закрыта). Нажимаем кнопку ОК. Всё, тег привязан. Таким образом, привязываем теги к остальным объектам на мнемосхеме. Рисунок 2.20 показывает, что все нужные теги подсоединены и успешно показывают текущее состояние объектов на мнемосхеме.

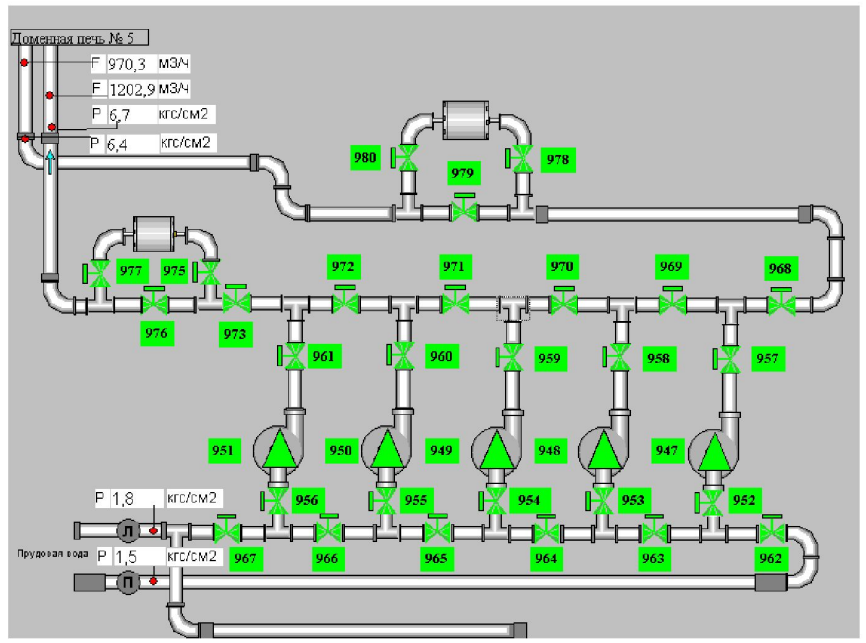


Рисунок 2.20 – Мнемосхема технологического объекта “Повысительная насосная станция с фильтровальной установкой” со всеми включенными объектами

РАЗДЕЛ 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ХРАНЕНИЯ ДАННЫХ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА

3.1. Разработка базы данных средствами СУБД MySQL

MySQL - это популярный сервер баз данных, используемый в разных приложениях. SQL означает язык структурированных запросов - (S)tructured (Q)uery (L)anguage, который MySQL использует для коммуникации с другими программами. Сверх того, MySQL имеет свои собственные расширенные функции SQL для того чтобы обеспечить пользователям дополнительный функционал. В этом документе мы рассмотрим как провести первоначальную установку MySQL, настроить базы данных и таблицы, и создать новых пользователей.

MySQL портирована на большое количество платформ: Linux, Windows, Mac OS, OpenServer и так далее.

Благодаря своей проверенной производительности, надежности и простоте использования база данных MySQL наиболее часто используется для веб-приложений на таких ресурсах, как Facebook, Twitter, YouTube и все пять из пяти лучших веб-сайтов. Кроме того, ее особенно часто выбирают в качестве встроенной базы данных, распространяемой тысячами поставщиков программного обеспечения и производителей оборудования.

Помимо языка MySQL, также необходимо иметь локальный сервер для проверки работоспособности баз данных, созданных с помощью MySQL. В качестве локального сервера был выбран Denwer.

Денвер (Denwer) – это набор программ для создания сайта на локальном компьютере, без выхода в Интернет.

Основные возможности программы Denwer.

Наверняка, многие разработчики веб-сайтов и/или баз данных сталкивались с проблемой, что при их создании часто сайт просто висит в Интернете, не принося никакого дохода. Сами понимаете, что хочется сначала полностью сконфигурировать сайт полностью, добавив туда все необходимые элементы типа текста, изображений, аудио или видео, а только потом выкладывать готовый продукт в Интернет. Именно таким конструктором и является программная среда Denwer.

Собственно, скачать локальный сервер Denwer или, как его еще называют, localhost можно на сайте разработчика. Там же есть подробная инструкция по установке и настройке программы на русском языке. Сама установка занимает всего несколько минут. Что касается конфигурации, сначала устанавливается сам сервер Denwer, затем создается собственная база данных, после чего активируется сайт, то есть на сервер «заливается» сам конструктор.

Созданный таким образом локальный сервер в программной среде Denwer, практически ничем не отличается от реального хостинга. По сути, он имеет те же компоненты: базы данных MySQL, сервер Apache, поддержку PHP, скрипты для работы с DB PhpMyAdmin и т.д. Кроме того, платформа Denwer позволяет использовать множество дополнительных плагинов и шаблонов. Однако, для их корректной работы для начала стоит их установить, протестировать и убедиться в работоспособности.

Для создание базы данных, запускаем Denwer, далее переходим по адресу localhost/tools/phpmyadmin в браузере. Откроется страница phpMyAdmin, где можно создавать базы данных на языке MySQL. Интерфейс phpMyAdmin представлен на рисунке 3.1.

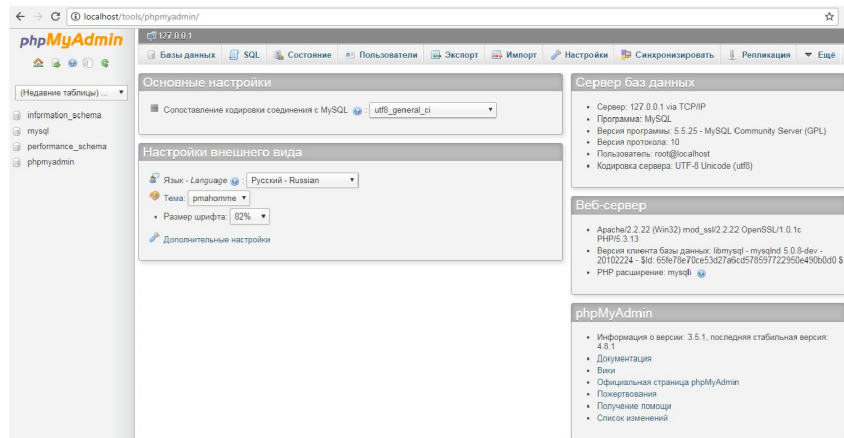


Рисунок 3.1 – Интерфейс PhpMyAdmin

Выбираем вкладку SQL, вводим следующий код для создания базы данных:

```
CREATE DATABASE station;
```

База данных создавалась. Создаём таблицы nasos и zadv средствами MySQL и phpMyAdmin. Таблицы имеют четыре поля:

- ID – тип поля integer(4);
- Message – тип поля varchar(20);
- Data – тип поля datetime;
- Status – тип поля boolean.

На рисунке 3.2 изображена структура таблицы zadv.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id_zadv	int(4)			Нет	Нет	AUTO_INCREMENT	Изменить
<input type="checkbox"/>	2 data	datetime			Нет	Нет		Изменить
<input type="checkbox"/>	3 message	varchar(20)	utf8_general_ci		Нет	Нет		Изменить
<input type="checkbox"/>	4 status	tinyint(1)			Нет	Нет		Изменить

Рисунок 3.2 – Структура таблицы zadv

3.2. Разработка программы для пересылки данных технологических параметров в БД

Связь с базой данных MySQL и WinCC возможна с помощью глобального сценария (Global Scripts).

Глобальные Сценарии (Global Scripts) есть не что иное, как общее определение функций и процедур Си, которое можно использовать в проекте или между проектами, в зависимости от их типа.

С помощью Global Scripts выполняются задачи, которые не зависят от того какой из экранов отображается в данный момент и отображается ли вообще какой-либо экран. Также Global Scripts используются для хранения глобальных функций и подпрограмм, доступных для вызова из других скриптов, как глобальных, так и привязанных к экранам.

Для динамизации свойств объекта используются С-макросы. С их помощью можно определить динамику свойства объекта в системе исполнения в зависимости от триггера, тега или состояния другого объекта. С-макросы следует использовать в том случае, если возможности, предоставляемые соединением с тегом или динамическим диалогом, не достаточны для решения вашей задачи.

С-макросы также можно использовать для программирования реакции на событие, произошедшее с графическим объектом. С-макрос следует использовать, если возможности, предоставляемые прямым соединением с тегом, не достаточны для решения вашей задачи. Использование макросов для определения реакции на изменение свойства объекта влияет на производительность системы исполнения. Событие происходит, если изменяется значение свойства объекта. В этом случае, макрос, связанный с этим событием, начинает выполняться. Когда кадр закрывается, выполнение всех макросов по очереди завершается. Это может привести к большой нагрузке на систему.

В графическом редакторе свойства объектов можно сделать динамическими при помощи процедур Си. Кроме того, процедуры Си могут использоваться для обработки событий объекта.

Редактор процедур.

Для конфигурирования процедур Си используется специальный редактор процедур. Данный редактор запускается из диалогового окна свойств объекта (Object Properties) нажатием правой кнопки мыши R на нужном свойстве или событии и последующим выбором процедуры Си во всплывающем меню. Уже существующие процедуры Си обозначаются зеленой стрелкой напротив свойства или события (см. рисунок 3.3).

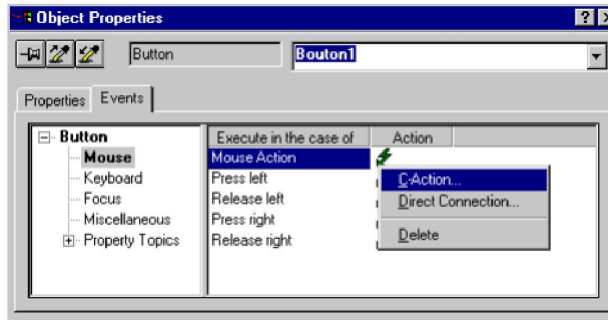


Рисунок 3.3 - Свойства объекта, у которого уже есть процедура Си

В редакторе процедур можно осуществлять программирование процедур Си. Для процедур Си, связанных со свойствами объекта, следует задавать триггер. Для процедур Си, связанных с событиями это делать не обязательно, т.к. событие само по себе является триггером. Готовые процедуры Си необходимо компилировать. Если компилятор не обнаружит ошибок, редактор процедур можно закрыть нажатием на кнопку ОК(см. рисунок 3.4).

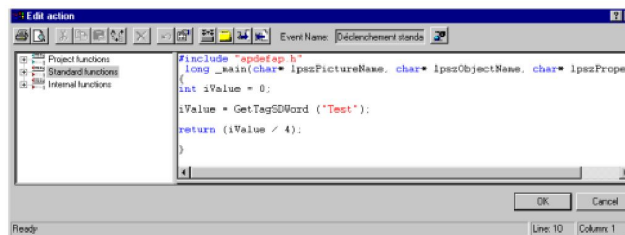


Рисунок 3.4 – окно редактора Си кода

Для связи базы данных MySQL с WinCC, создаём функцию `mysql_query_wincc`:

```

int mysql_query_wincc(char* query)
{
#include "apdefap.h"
#pragma code("libmysql.dll")
#include "mysql.h"
#pragma code()
MYSQL *conn;
MYSQL_RES *results;
conn = mysql_init(NULL);
if (!mysql_real_connect(conn,"127.0.0.1","admin","admin","station",0,
NULL,0))
{
printf("%s\r\n", mysql_error(conn));
exit(0);
}
mysql_query(conn, query);
mysql_close(conn);
printf(query);
printf("\r\n");
return 0;}

```

Эту функцию создаём в разделе Project functions (функции проекта). После

её создания, необходимо её откомпилировать и сохранить (см. рисунок 3.5).

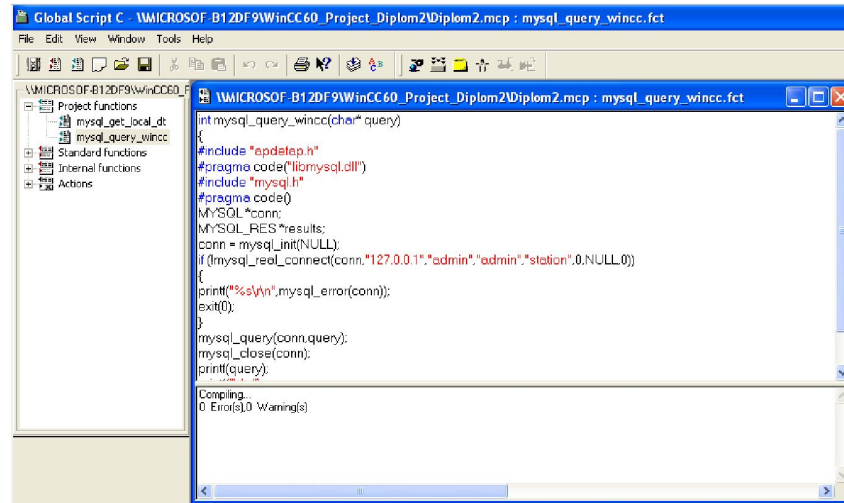


Рисунок 3.5 – Функция подключения к базе данных MySQL

Для занесения изменений в базу данных, необходимо также прописать скрипт определённому объекту на мнемосхеме. Открываем графический дизайнер, далее нажимаем правой кнопкой мыши на нужный объект. Выбираем Properties(свойства). После открытия окна свойств, выбираем пункт Colors и кликаем правой кнопкой мыши на лампочке рядом со строчкой Background Color и выбираем C-Action. Открылось окно редактирования скрипта на языке Си. Создаём скрипт для занесения значений в базу данных:

```
#include "apdefap.h"

long _main(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
#define TAG_1 "zadv"
char query[2600]=""; // Объявление символьной переменной длиной 2600
символов.
```

```

if ((unsigned long)GetTagDouble (TAG_1)) // изменение цвета индикатора
    return 65280;
else
    return 255;
if (GetTagBit("zadv") == TRUE) // Проверка тега WinCC на выполнение
условия.
    sprintf (query,"INSERT INTO `zadv` (`data`,`message`,`status`) VALUES
('%s', '%c', '%s'",
            mysql_get_local_dt( "%u-%u-%u %u:%u:%u"),
            "Задвижка 1",
            GetTagBit("Zadv")
            ); // Наполнение SQL-запроса значениями.
mysql_query_wincc(query);
return 0;
}

```

С помощью данного кода, в базу данных при каждом изменении параметра объекта будут передаваться данные (см. рисунок 3.6).

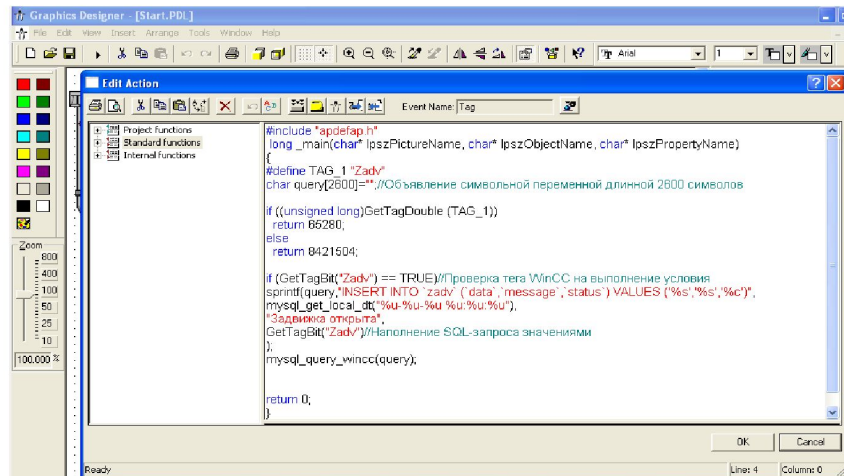


Рисунок 3.6 – Скрипт для передачи данных в базу данных MySQL

Следующий рисунок показывает данные, которые занесли после выполнения действий с задвижками.

		id_zadv	data	message	status
<input type="checkbox"/>	Изменить Копировать Удалить	1	2018-06-03 14:13:07	Задвижка 1	1
<input type="checkbox"/>	Изменить Копировать Удалить	2	2018-06-11 12:09:00	Задвижка 5	1
<input type="checkbox"/>	Изменить Копировать Удалить	3	2018-06-11 12:10:00	Задвижка 3	0
<input type="checkbox"/>	Изменить Копировать Удалить	4	2018-06-11 12:14:00	Задвижка 8	1
<input type="checkbox"/>	Изменить Копировать Удалить	5	2018-06-11 12:21:00	Задвижка 9	1
<input type="checkbox"/>	Изменить Копировать Удалить	6	2018-06-11 12:25:00	Задвижка 20	1
<input type="checkbox"/>	Изменить Копировать Удалить	7	2018-06-11 12:27:00	Задвижка 15	1
<input type="checkbox"/>	Изменить Копировать Удалить	8	2018-06-11 12:39:00	Задвижка 12	0
<input type="checkbox"/>	Изменить Копировать Удалить	9	2018-06-11 12:40:00	Задвижка 10	1
<input type="checkbox"/>	Изменить Копировать Удалить	10	2018-06-11 12:45:00	Задвижка 23	1
<input type="checkbox"/>	Изменить Копировать Удалить	11	2018-06-11 12:46:00	Задвижка 14	0

Рисунок 3.7 – Итог выполнения скрипта

ЗАКЛЮЧЕНИЕ

Все необходимые цели дипломной работы были выполнены. В ходе проекта была реализована мнемосхема технологического объекта “Повысительная насосная станция с фильтровальной установкой”, разработана и подключена база данных MySQL.

В первом разделе рассмотрены теоретико-практические аспекты разработки мнемосхем визуализации технологического объекта, такие как общие принципы разработки мнемосхем при помощи SCADA-систем, обзор трёх SCADA-систем, критерии выбора SCADA-систем. Для разработки SCADA-системы была выбрана SCADA-система Simatic WinCC.

Во втором разделе разработана мнемосхема визуализации технологического объекта с помощью SCADA-системы, разработано отображение состояний механизмов и датчиков на технологических мнемосхемах, приведены технологические параметры мнемосхемы.

В третьем разделе реализовано программное хранение данных технологического процесса и для этой цели были создана база данных средствами системы управления базой данных MySQL и разработаны C-макросы для открытия базы данных и пересылки данных технологических параметров с объектов в базу данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое SCADA-система [Электронный ресурс]: – Режим доступа: <https://ru.wikipedia.org/wiki/SCADA> - SCADA
2. Что такое мнемосхема [Электронный ресурс]: – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%B5%D0%BC%D0%BE%D1%81%D1%85%D0%B5%D0%BC%D0%B0> - Мнемосхема
3. Мнемосхемы [Электронный ресурс]: – Режим доступа: http://www.ie.tusur.ru/books/COI/page_38.htm - Мнемосхемы
4. Обзор SCADA-системы RSView32 [Электронный ресурс]: – Режим доступа: <http://www.eskovostok.ru/catalog/software/capacity/rsview> - RSView - линейка продуктов программного обеспечения человеко-машинного интерфейса
5. Обзор SCADA-системы SIMATIC WinCC [Электронный ресурс]: – Режим доступа: <http://www.energostandart.ru/tehnika-avtomatisacii/visualisacija/scada-sistema-vizualizacii-simatic-wincc/> - SCADA система визуализации WinCC.
6. Обзор SCADA-системы Wonderware InTouch [Электронный ресурс]: – Режим доступа: <https://www.wonderware.ru/hmi-scada/intouch/> - InTouch Wonderware Russia
7. Критерии выбора SCADA-системы [Электронный ресурс]: – Режим доступа: <http://a-sys.com.ua/helpful/vybor-scada-sistemy.html> - Выбор SCADA-системы для разработки верхнего уровня АСУ ТП
8. Теплоэнергетика металлургического производства: Учебное пособие./ Б.И. Басок, Ю.Л. Курбатов, А.Б. Бирюков, Е.В. Новикова: Донецк ГБУЗ «ДонНТУ» 2013 – 25 с.
9. Палитры в графическом редакторе WinCC [Электронный ресурс]: – Режим доступа: https://cache.industry.siemens.com/dl/files/587/1144587/att_34787/v1/GraphicsDesigner_V502_r.pdf - WinCC графический редактор, с. 9-36.

ПРИЛОЖЕНИЕ А

ИЛЛЮСТРАЦИИ ДИПЛОМНОЙ РАБОТЫ

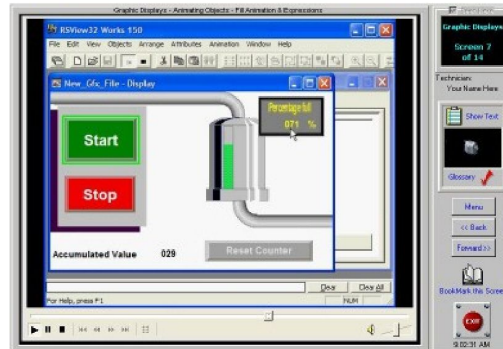


Рисунок 1 – Проект, созданный в системе RSView32



Рисунок 2 – Проект, созданный в системе SIMATIC WinCC

ПРИЛОЖЕНИЕ А (продолжение)

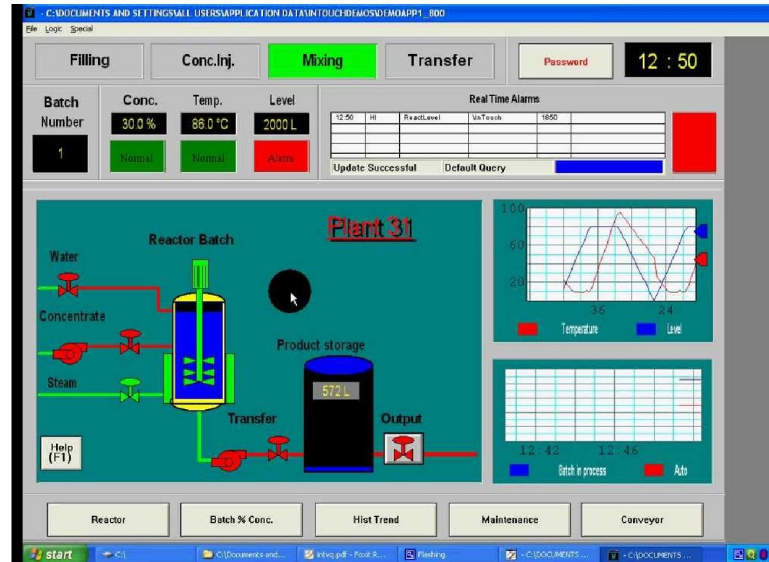
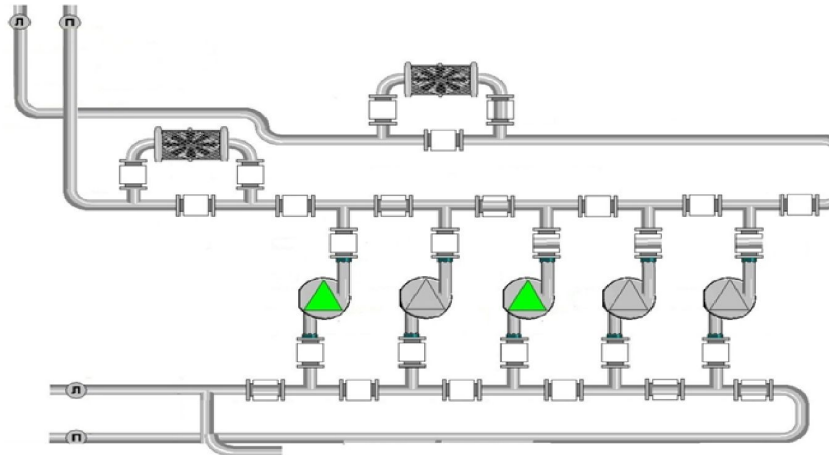


Рисунок 3 – Проект, созданный в системе Wonderware InTouch



												Лист
										EEEE 000000 000 000 ПП		4
Изм.	Лист	№ докум.	Подпис	Дата								

Рисунок 4 – Схема повысительной насосной станции с фильтровальной установкой

ПРИЛОЖЕНИЕ А (продолжение)

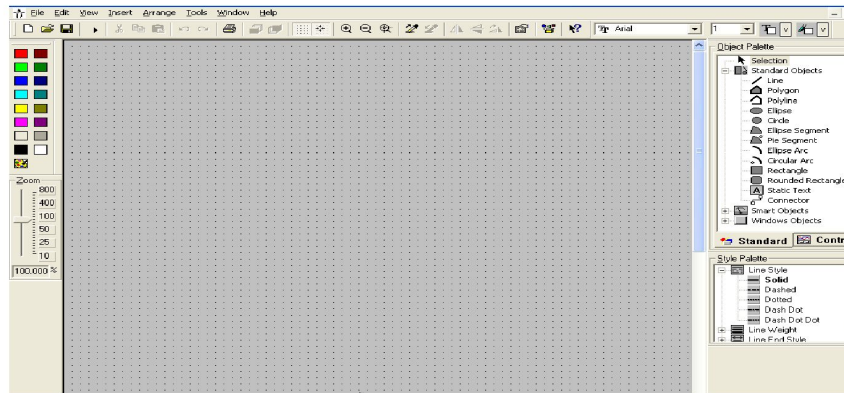


Рисунок 5 – Окно редактора кадра



Рисунок 6 – Палитра цветов

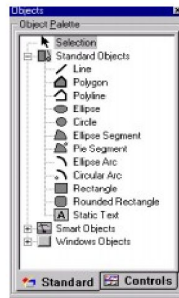


Рисунок 7 – Палитра объектов

ПРИЛОЖЕНИЕ А (продолжение)



Рисунок 8 – палитра стилей



Рисунок 9 – Палитра масштабирования

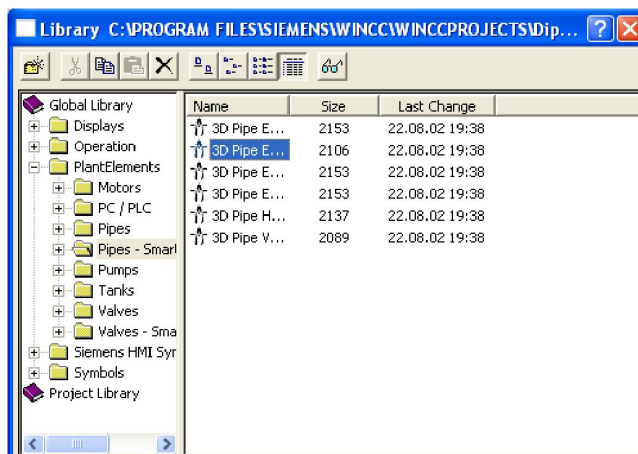


Рисунок 10 – Библиотека объектов

ПРИЛОЖЕНИЕ А (продолжение)

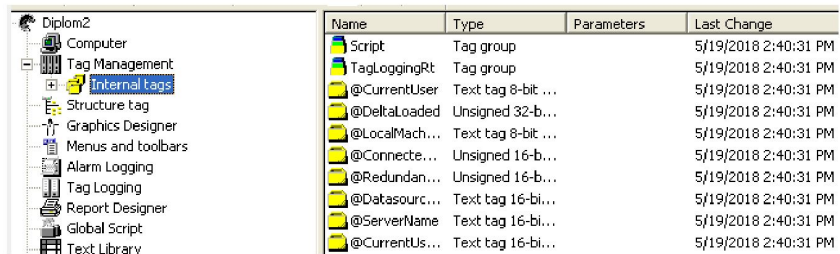


Рисунок 11 – Внутренние теги проекта

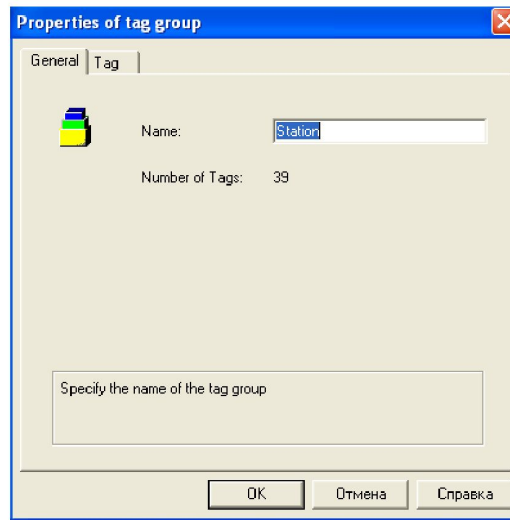


Рисунок 12 – Окно свойств группы тегов

ПРИЛОЖЕНИЕ А (продолжение)

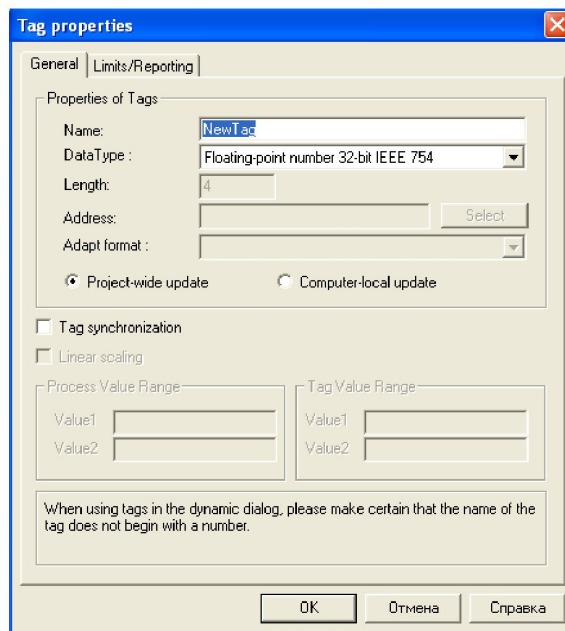


Рисунок 13 – Окно свойств тега

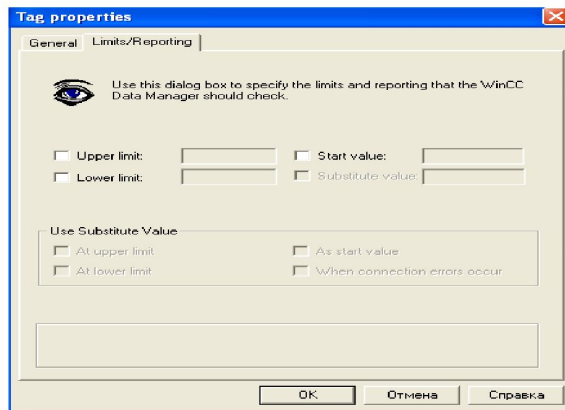


Рисунок 14 – вкладка Limits/Reporting окна Tag properties

ПРИЛОЖЕНИЕ А (продолжение)

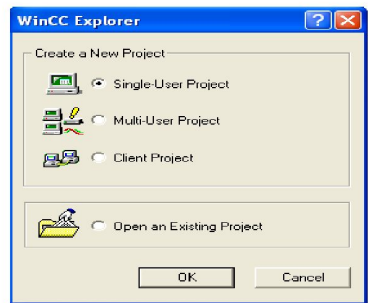


Рисунок 15 – Варианты создания проекта

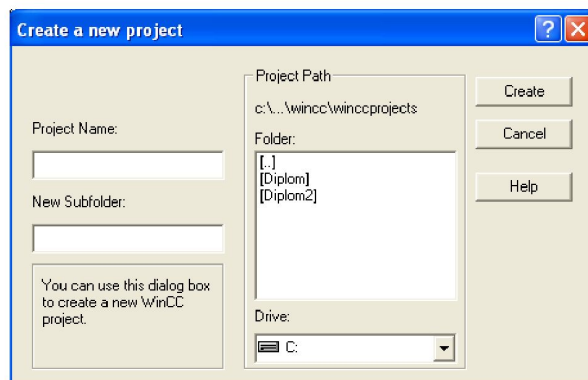


Рисунок 16 – Окно ввода названия проекта и его создания

ПРИЛОЖЕНИЕ А (продолжение)

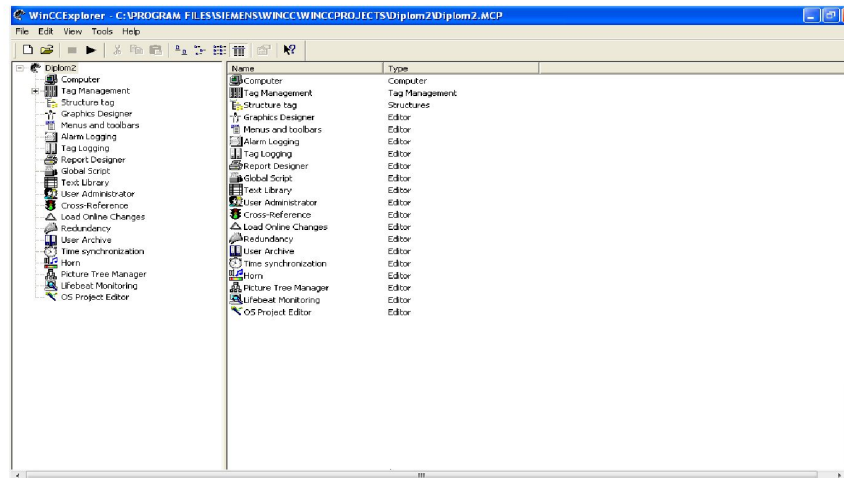


Рисунок 16 – Окно с компонентами и настройками проекта

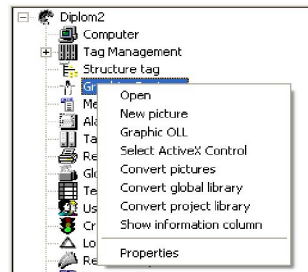


Рисунок 17 – Пункты меню при создании нового кадра процесса

ПРИЛОЖЕНИЕ А (продолжение)

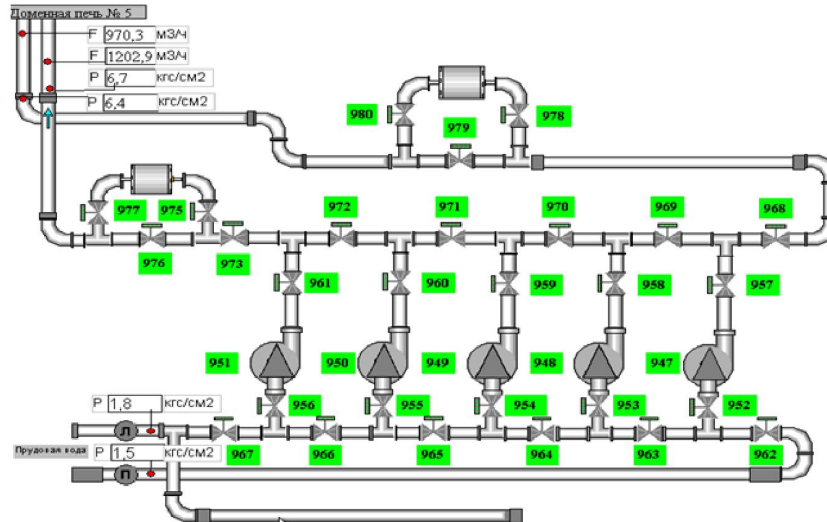


Рисунок 18 – Мнемосхема технологического объекта “Повысительная насосная станция с фильтровальной установкой” без включенных объектов

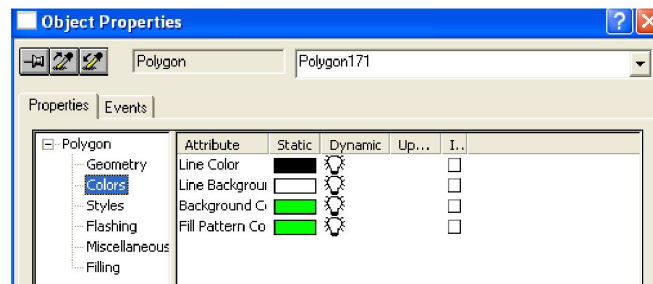


Рисунок 19 – Окно свойств объекта

ПРИЛОЖЕНИЕ А (продолжение)

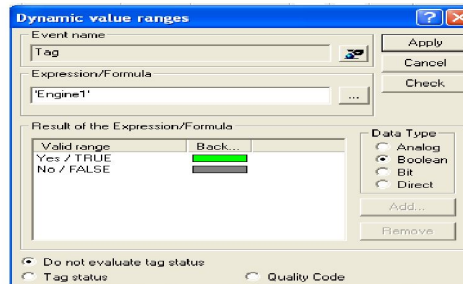


Рисунок 20 – окно Dynamic Dialog

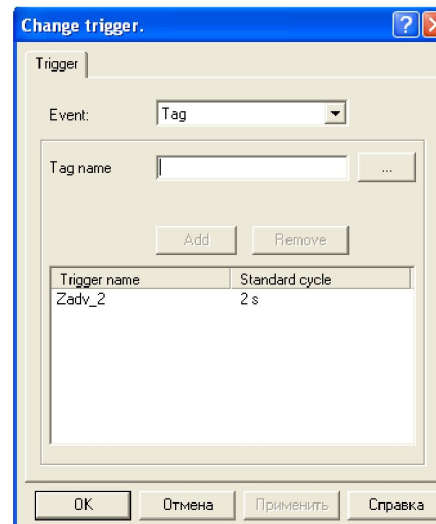


Рисунок 21 – окно изменения события.

ПРИЛОЖЕНИЕ А (продолжение)

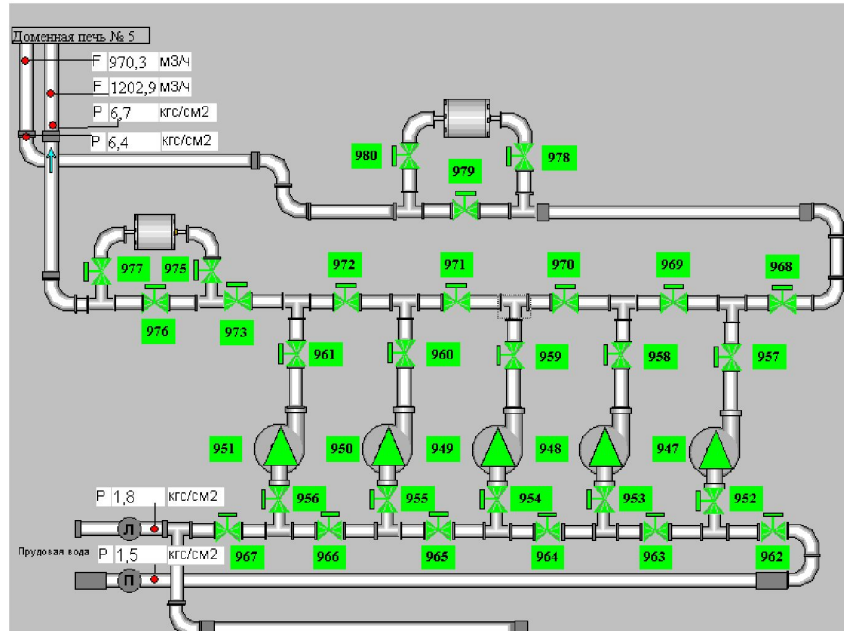


Рисунок 22 – Мнемосхема технологического объекта “Повысительная насосная станция с фильтровальной установкой” со всеми включенными объектами

ПРИЛОЖЕНИЕ А (продолжение)

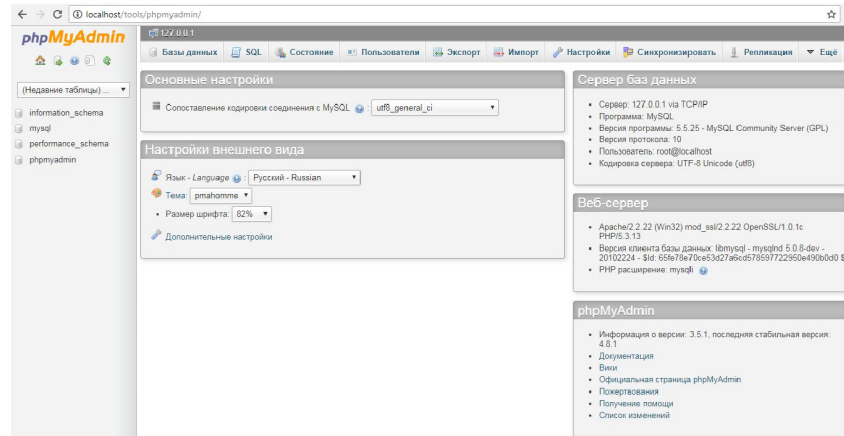


Рисунок 23 – Интерфейс PhpMyAdmin

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id_zadv	int(10)			Нет	Нет	AUTO_INCREMENT	Изменить
<input type="checkbox"/>	2 data	datetime			Нет	Нет		Изменить
<input type="checkbox"/>	3 message	varchar(20)	utf8_general_ci		Нет	Нет		Изменить
<input type="checkbox"/>	4 status	tinyint(1)			Нет	Нет		Изменить

Рисунок 24 – Структура таблицы zadv

ПРИЛОЖЕНИЕ А (продолжение)

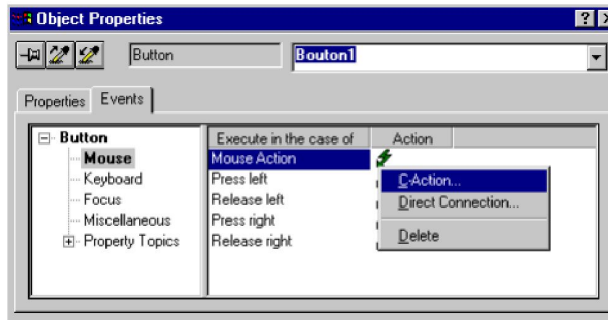


Рисунок 25 - Свойства объекта, у которого уже есть процедура Си

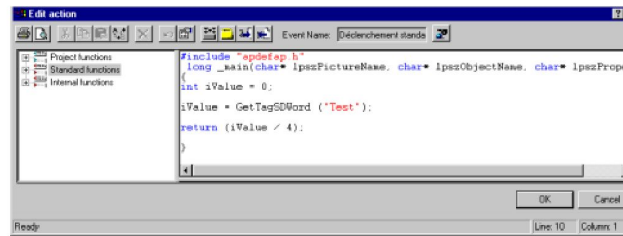
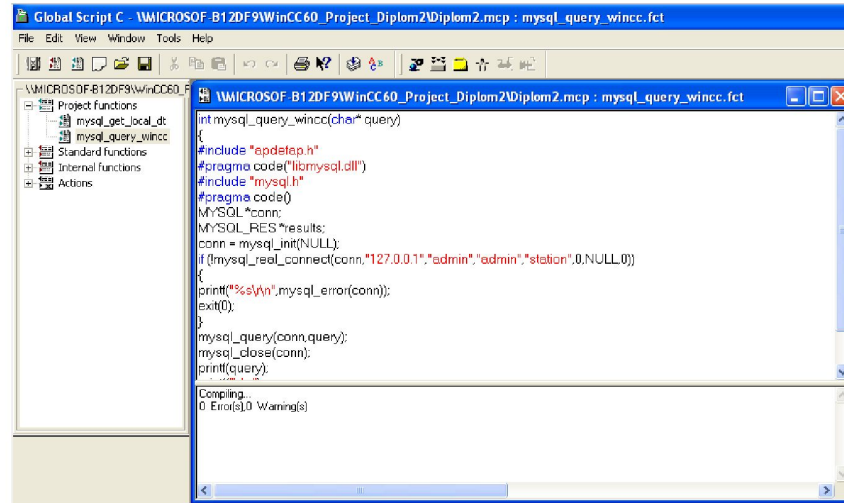


Рисунок 26 – окно редактора Си кода

ПРИЛОЖЕНИЕ А (продолжение)



The screenshot shows a window titled "Global Script C - \\MICROSOFT\B12DF9WinCC60_Project_Diplom2\Diplom2.mcp : mysql_query_wince.fct". The editor contains the following code:

```
int mysql_query_wince(char* query)
{
#include "epdefep.h"
#pragma code("libmysql.dll")
#include "mysql.h"
#pragma code()
MYSQL* conn;
MYSQL_RES *results;
conn = mysql_init(NULL);
if (!mysql_real_connect(conn,"127.0.0.1","admin","admin","station",0,NULL,0))
{
printf("%s\n",mysql_error(conn));
exit(0);
}
mysql_query(conn,query);
mysql_close(conn);
printf(query);
}

Compiling...
0 Error(s)0 Warning(s)
```

Рисунок 27 – Функция подключения к базе данных MySQL

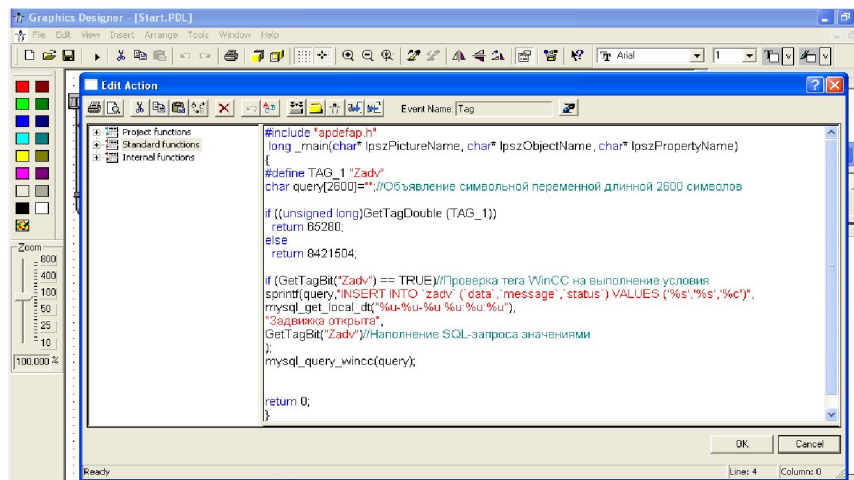


Рисунок 28 – Скрипт для передачи данных в базу данных MySQL

ПРИЛОЖЕНИЕ А (продолжение)

				id_zadv	data	message	status			
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	2018-06-03 14:13:07	Задвижка 1	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	2018-06-11 12:09:00	Задвижка 5	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	2018-06-11 12:10:00	Задвижка 3	0
<input type="checkbox"/>		Изменить		Копировать		Удалить	4	2018-06-11 12:14:00	Задвижка 8	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	5	2018-06-11 12:21:00	Задвижка 9	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	6	2018-06-11 12:25:00	Задвижка 20	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	7	2018-06-11 12:27:00	Задвижка 15	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	8	2018-06-11 12:39:00	Задвижка 12	0
<input type="checkbox"/>		Изменить		Копировать		Удалить	9	2018-06-11 12:40:00	Задвижка 10	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	10	2018-06-11 12:45:00	Задвижка 23	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	11	2018-06-11 12:46:00	Задвижка 14	0

Рисунок 29 – Итог выполнения скрипта

ПРИЛОЖЕНИЕ Б
СЦЕНАРИИ, ИСПОЛЬЗУЕМЫЕ ПРИ ОДКЛЮЧЕНИИ БАЗЫ
ДАННЫХ

Подключение к базе данных.

```
int mysql_query_wincc(char* query)
{
#include "apdefap.h"
#pragma code("libmysql.dll")
#include "mysql.h"
#pragma code()
MYSQL *conn;
MYSQL_RES *results;
conn = mysql_init(NULL);
if (!mysql_real_connect(conn,"127.0.0.1","admin","admin","station",0,
```

					EEEE 000000 000 000 ПП	Лист
Изм.	Лист	№ докум.	Подпис	Дата		4

```

NULL,0))
{
printf("%s\r\n", mysql_error(conn));
exit(0);
}
mysql_query(conn, query);
mysql_close(conn);
printf(query);
printf("\r\n");
return 0;
}

```

Запись данных в базу данных.

```
#include "apdefap.h"
```

```
long _main(char* lpszPictureName, char* lpszObjectName, char*
```

ПРИЛОЖЕНИЕ Б (продолжение)

```
lpszPropertyName)
```

```
{
```

```
#define TAG_1 "zadv"
```

```
char query[2600]=""; // Объявление символьной переменной длиной 2600
```

символов.

```
if ((unsigned long)GetTagDouble (TAG_1)) // изменение цвета индикатора
```

```
return 65280;
```

```
else
```

```
return 255;
```

if (GetTagBit("zadv") == TRUE) // Проверка тега WinCC на выполнение условия.

```
sprintf (query,"INSERT INTO `zadv` (`data`,`message`,`status`) VALUES
```

```
(%s','%s','%c)',
mysql_get_local_dt( "%u-%u-%u %u:%u:%u"),
"Задвижка 1",
GetTagBit("Zadv")
); // Наполнение SQL-запроса значениями.
mysql_query_wincc(query);
return 0;
}
```


те

Обозначение документа	Условное	мечания	Содержание	Подпись студента	Подпись
			тъекта й» элера:		