

А.А. Силаев, Е.Ю. Силаева, М.А. Трушников

**Интегрированные системы
проектирования и управления
автоматизированных и автоматических
производств**



Волжский
2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ВОЛЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

А.А. Силаев, Е.Ю. Силаева, М.А. Трушников

**Интегрированные системы
проектирования и управления
автоматизированных и автоматических
производств**

Электронное учебное пособие



2018

УДК: 658.52(075.5)

ББК 32.9

С 36

Рецензенты:

кандидат физико-математических наук, доцент кафедры «Автоматизация технологических процессов и производств» филиала НИУ МЭИ в г. Волжском

Е.В. Капля,

кандидат технических наук, доцент кафедры «Автоматизации производственных процессов» ВолгГТУ

А.Г. Алёхин

Печатается по решению редакционно-издательского совета
Волгоградского государственного технического университета

Силаев, А.А.

Интегрированные системы проектирования и управления автоматизированных и автоматических производств [Электронный ресурс] : учебное пособие / А. А. Силаев, Е.Ю. Силаева, М.А. Трушников; ВПИ (филиал) ВолгГТУ. – Электрон. текстовые дан. (1 файл: 1,97МБ). – Волжский, 2018. – Режим доступа: <http://lib.volpi.ru>. – Загл. с титул. экрана.

ISBN 978-5-9948-2772-7

В пособие рассмотрена интегрированная информационная система управления предприятием с подразделением на уровни. Наибольшее внимание уделено уровню автоматизации технологических процессов и производств. Учебно-методическое пособие представляет собой конспект лекций.

Предназначено для студентов технических ВУЗов обучающихся по направлению магистратуры «Автоматизация технологических процессов и производств» всех форм обучения.

Ил. 16, табл. 1, библиограф.: 20 назв.

ISBN 978-5-9948-2772-7

© Волгоградский государственный
технический университет, 2018

© Волжский политехнический
институт, 2018

Содержание

Введение	4
1 Программное обеспечение интегрированных систем	6
1.1 Некоторые проблемы интеграции	7
1.2 OLAP-системы	10
1.3 ERP-системы	12
1.4 MES-системы	14
2 SCADA-системы	17
2.1 Характеристики SCADA-систем	17
2.2 Основные компоненты SCADA-систем	20
2.3 Языки программирования SCADA-систем	22
2.3.1 Язык FBD	24
2.3.2 Язык LD	27
2.3.3 Язык SFC	30
2.3.4 Язык ST	34
2.3.5 Язык IL	37
3 Обзор SCADA-систем, представленных на рынке	40
3.1 SCADA-система InTouch	40
3.2 SCADA-система Genesis32	44
3.3 Программный продукт TIA Portal	46
3.3.1 SCADA-система WinCC	46
3.3.2 Дополнительный набор опций WinCC	48
3.4 SCADA-система CitectSCADA	52
3.5 SCADA-система Trace Mode	58
3.6 SCADA-система MasterSCADA	63
Заключение	69
Список литературы	71

Введение

Применение автоматизированных систем управления является ключевым направлением для повышения эффективности производств в различных сферах.

Современные автоматизированные системы проектируются как единые интегрированные системы, объединяющие всю деятельность предприятия в целом. Системы управления используют сложные алгоритмы обработки информации, которую получают как путём ручного ввода, так и с помощью автоматических измерений.

К обработке информации относят:

- автоматизированный сбор,
- преобразование,
- хранение и передачу,
- визуализацию.

В результате синтезируется информация, на основании которой система в автоматическом или автоматизированном режиме управляет технологическими процессами или производством в целом.

Такие системы обладают высокой оперативностью и эффективностью. В них сведён до минимума человеческий фактор. Эти системы принято называть интегрированными системами управления. Но стоит отметить, что внедрение подобных систем трудоёмкий процесс, в результате которого перестраиваются не только системы управления предприятием, но и сами процессы управления. Это требует поддержки со стороны руководства предприятия.

Учебное пособие представляет собой конспект лекций.

Данное учебное пособие предназначено для студентов магистратуры, обучающихся по направлению «Автоматизация технологических процессов и производств» при изучении дисциплины «Интегрированные

системы проектирования и управления автоматизированных и автоматических производств» всех форм обучения.

1 Программное обеспечение интегрированных систем

Комплексную (корпоративную) информационную систему (КИС) предприятия в общем случае можно представить в виде иерархии уровней, охватывающих все виды организационно-управленческой деятельности на предприятии, от сбора данных о технологических процессах до контроля выполнения заказов[18]. На рисунке 1 схематично показана структура информационной системы предприятия.



Рисунок 1 – Структура информационной системы предприятия.

В общем, информационную структуру предприятия можно разделить на два уровня:

верхний уровень — это уровень управления бизнес-процессами предприятия в целом;

нижний уровень — это уровень управления непосредственно технологическими процессами.

Промежуточным звеном между уровнями являются SCADA системы, которые позволяют объединить все процессы предприятия. Поэтому основное

внимание в пособие уделим именно им. Но предварительно кратко рассмотрим все уровни информационной системы предприятия. При этом следует отметить, что деление на уровни достаточно условно, и смежные системы могут частично выполнять функции друг друга. Например, MES-системы могут получать данные от датчиков, также как и SCADA-системы.

1.1 Некоторые проблемы интеграции

Прежде чем перейти к рассмотрению уровней информационной системы предприятия, уделим внимание некоторым проблемам, которые возникают у российских предприятий при интеграции всей системы управления производством.

Традиционно внедрением SCADA-систем в России занимаются фирмы-интеграторы, которые на базе разработок своих партнёров, производителей программного и аппаратного обеспечения, создают готовые решения для конечных потребителей. Иногда конкуренцию им составляют ИТ-отделы предприятий. При этом используется различное программно-аппаратное обеспечение российских и зарубежных производителей. Достаточно часто задачи решаются локально, без системного подхода и учёта требований к дальнейшей интеграции этих подсистем в информационную систему предприятия.

Многие известные SCADA-системы содержат реализацию всех необходимых функций оперативного контроля и управления технологическим процессом. Но наблюдаются ситуации, когда использование того или иного SCADA-пакета вызывает проблемы при интеграции с другими системами.

Одна из таких проблем – отсутствие в SCADA-системе модулей, ответственных за агрегацию данных из разнородных источников (самой SCADA-системы, OPC, SQL-базы и ряда других источников). Без наличия подобного функционала невозможно будет подготовить и передать данные в MES- и ERP-системы. При этом необходимо поддерживать высокоуровневое

пространство имён (формирование полного имени переменной, например, завод-цех-линия-станок-параметр) и организацию доступа к данным через системы управления базами данных. В этом случае системы MES и ERP посредством обычных SQL-запросов к базе данных смогут легко получить доступ к истории и оперативным данным с уровня SCADA-системы.

Другая проблема интеграции внутри информационной системы предприятия – неоднородность информационных потоков в современных диспетчерских системах. Среди информационных потоков можно выделить два типа: технологические обмены и обмены бизнес-информацией.

Первые – это обмены в реальном времени значениями измеренных и контролируемых параметров. Такие обмены касаются SCADA-систем и осуществляются диспетчерскими пунктами с локальными системами автоматизации, а также идут между диспетчерскими пунктами.

Бизнес-обмены – это обмены во временном режиме процессов хозяйственной деятельности показателями, касающимися производственной деятельности предприятия. Это обмены данными в основном между компонентами MES-системы с ERP-системой, обмены с системами автоматизации внешних организаций.

Однако основной сложностью в организации информационного обмена между нижним и верхним уровнем КИС является однозначное определение перечней сигналов для обмена, согласование логик, кодировок, состояния и других вопросов. Как для технологических обменов, так и для обменов бизнес-информацией перечисленные вопросы могут привести к большим проблемам. Для технологических обменов это, прежде всего, традиционная несогласованность логик, кодировок, представления информации для отдельных установок, объектов, диспетчерских комплексов локального уровня и корпоративного. Даже на уровне установки и объекта один и тот же параметр может иметь различное наименование, например, по причине поставки

разными производителями SCADA-системы и локальной системы автоматизации.

Ещё более серьёзной проблемой является определение соответствия между параметрами и объектами в MES- и ERP-системах. В силу большого числа причин в ряде случаев имеются расхождения в определении (и описании) объекта предприятия в бухгалтерских документах и в диспетчерской отчётности, не говоря уже о привязке значения к средствам измерения и контроля в SCADA-системе. Большую путаницу вносят ремонты, замены оборудования. Имеющиеся разночтения приводят к существенному усложнению интеграционных проектов и повышают затраты как на внедрение системы, так и на её сопровождение и расширение. Для решения этой проблемы предлагается однозначное определение места источника данных в иерархии предприятия. Источник данных может изменяться (а также добавляться, удаляться) только в рамках реконструкции всего предприятия, а не ремонтов или замены технологического оборудования (то есть, например, замена задвижки в результате ремонта не должна приводить к изменению кодировок параметров давления и температуры, измеряемых на данной задвижке). Для этих целей предлагается введение паспортизации параметров, выполняемое ещё на этапе проектирования.

Также зачастую при попытке заказчика получить целостную КИС интегратор сталкивается со следующей проблемой – неготовность предприятия к внедрению MES- и ERP-системы. Как бы ни хотелось, но практика показывает: чтобы на предприятии заработала стандартизованная система управления, оно должно вписываться в эти стандарты, то есть выполняемые бизнес-процессы должны удовлетворять ряду требований (имеются в виду требования стандартов MES- и ERP-систем).

Для целого ряда предприятий полный функционал бывает и не нужен, даже в продуктах от грандов систем управления предприятием, таких как SAP, Microsoft, Oracle и других, используются далеко не все модули.

В целом можно сказать, что основа эффективной организации информационных обменов – полная проработка данных вопросов на этапе проектных работ. Роль проектных организаций крайне велика, в случае эффективного решения проблемы информационных обменов на стадии проектирования достигается существенная экономия всех видов ресурсов при реализации и расширении системы и, главное, обеспечивается эффективное решение всех классов задач на всех уровнях управления[18].

1.2 OLAP-системы

OLAP (OnLine Analytical Processing – интерактивная аналитическая обработка) – это технология обработки данных, заключающаяся в подготовке обобщённой информации на основе больших массивов данных, структурированных по многомерному принципу. Реализации технологии OLAP являются компонентами программных решений задач управления бизнес-процессами [13].

Принятие руководителем решений по управлению предприятием попадает в область задач наиболее сложно поддающихся автоматизации. Поэтому в основном используются системы поддержки принятия решения.

OLAP-системы обладают средствами визуализации данных в удобном виде для восприятия и анализа пользователем. В итоге образуются многомерный набор данных — OLAP-куб, оси которого содержат наборы факторов, а ячейки – значения полученные при конкретных условиях. Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации.

По измерениям в многомерной модели откладывают факторы, влияющие на деятельность предприятия. Полученный OLAP-куб затем наполняется показателями деятельности предприятия (цены, продажи, план, прибыли, убытки и т.п.). Наполнение куба может вестись как реальными данными, так и прогнозируемыми на основе исторических данных.

В процессе анализа пользователь может менять точку зрения на данные (так называемая операция смены логического взгляда), тем самым, просматривая данные в различных разрезах и разрешая конкретные задачи. Над кубами могут выполняться различные операции, включая прогнозирование и условное планирование (анализ типа “что, если”). Пример OLAP-куба приведён на рисунке 2.

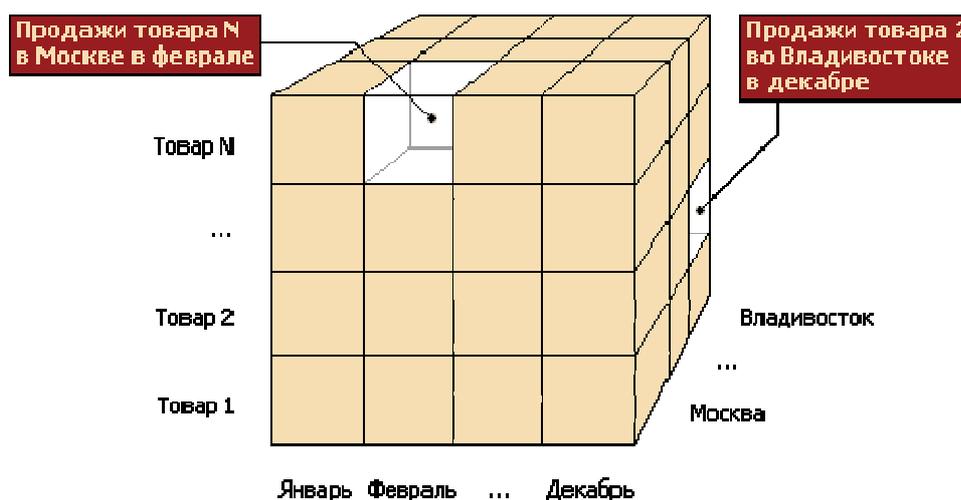


Рисунок 2 – Пример OLAP-куба.

Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных. Оперативная аналитическая обработка позволяет значительно упростить и ускорить процесс подготовки и принятия решений руководящим персоналом. Оперативная аналитическая обработка служит цели превращения данных в информацию. Она принципиально отличается от традиционного процесса поддержки принятия решений, основанного, чаще всего, на рассмотрении структурированных отчетов.

Программные продукты: Microsoft SQL Server Analysis Services, Hyperion Essbase, Cognos PowerPlay, BusinessObjects, MicroStrategy, SAP BW.

Для решения простых задач анализа данных возможно использовать офисные приложения Excel и Access компании Microsoft, которые содержат элементарные средства OLAP-технологии, позволяющие создавать сводные таблицы и строить на их основе различные отчеты.

1.3 ERP-системы

ERP (Enterprise-Resource Planning – планирование ресурсов предприятия) – комплексная система управления ресурсами предприятия, объединяющая её подразделения и отражающая их функции. ERP-системы используются для реализации различных методов эффективного планирования всеми ресурсами на предприятии с целью осуществления производства, закупок, учета, продаж, выполнения заказов клиентов в сферах обслуживания.

На смену устаревшим локальным самописным компьютерным системам по финансам, контролю над производством, управлению персоналом, логистике, складу приходят универсальные ERP-системы, включающие в себя программные модули, позволяющие полностью сохранять функционал старых подсистем и вводить новые функциональные возможности, касающиеся сводной аналитической отчетности и планирования. При этом существенно упрощается и сокращается процесс обмена данными между различными подразделениями на предприятии. Основными трудностями внедрения таких систем являются нарушение нормального ритма работы подразделений в период внедрения и отладки, сложности при первичном наполнении исходными данными системы и переносе данных из устаревших программ, противодействие со стороны персонала, нежелающего менять привычный уклад работы [17].

В состав практически любой ERP-системы входит следующий набор модулей, обеспечивающих работу и взаимодействие всех основных подразделений предприятия:

- производство;
- снабжение;
- торговля и сбыт произведенной продукции;
- бухгалтерский и налоговый учет;
- управление персоналом;
- складское хранение;

- техобслуживание оборудования;
- управление и планирование финансов.

Обобщённая структура ERP-системы показана на рисунке 3.



Рисунок 3 – Обобщённая структура ERP-системы.

Основные функциональные задачи ERP-систем:

- регулировать количество запасов, устраняя их дефицит и залеживание на складах, и тем самым значительно снизить складские издержки;
- оценивать выполнимость поступивших заказов с точки зрения имеющихся на предприятии мощностей;
- сократить расходы и время, затрачиваемые на изготовление продукции, за счет оптимизации бизнес-процессов;

– отслеживать фактическую производительность каждой производственной единицы и, сравнивая ее с плановой производительностью, оперативно вносить корректировки в производственные планы;

– в результате уменьшения цикла производства и цикла выполнения заказа более гибко реагировать на запросы клиентов;

– улучшить обслуживание клиентов и заказчиков за счет своевременного исполнения заказов [9].

Как правило, ERP-система не связана непосредственно с производственным процессом, но имеет модель технологического процесса и получает необходимые данные от программ управления технологическим процессом нижнего уровня иерархии: MES- и SCADA-систем.

Работа ERP-системы состоит в улучшении деятельности предприятия, оптимизации материальных и финансовых потоков на основе вводимой на рабочих местах необходимой информации.

Внедрение ERP-систем позволяет достичь конкурентных преимуществ за счет оптимизации бизнес-процессов предприятия и снижения издержек, дает возможность управлять себестоимостью продукции. Наиболее известные ERP-системы и ERP-подобные системы, предлагаемые в настоящее время на рынке — это SAP R/3, Oracle Applications, 1С: Предприятие.

1.4 MES-системы

MES (Manufacturing Executive Systems – система управления производственными процессами) — это система, предназначенная для решения задач синхронизации, координации, анализа и оптимизации выпуска продукции в рамках какого-либо производства, отражающая производственный процесс и его полный производственный цикл.

Внедрение MES-систем позволяет решать актуальные для производства задачи (см. рисунок 4):

- активация производственных мощностей на основе детального планирования производства;
- отслеживание производственных мощностей;
- сбор информации, связанной с производством, от:
 - систем автоматизации производственного процесса,
 - датчиков,
 - оборудования,
 - персонала,
 - программных систем;
- отслеживание и контроль параметров качества;
- обеспечение персонала и оборудования информацией, необходимой для начала процесса производства;
- установление связей между персоналом и оборудованием в рамках производства;
- установление связей между производством и поставщиками, потребителями, инженерным отделом, отделом продаж и менеджментом.
- реагирование на:
 - требования по номенклатуре производства,
 - изменение компонентов, сырья и полуфабрикатов, применяемых в процессе производства,
 - изменение спецификации продуктов,
 - доступность персонала и производственных мощностей;
- соответствие индустриальным стандартам [10].

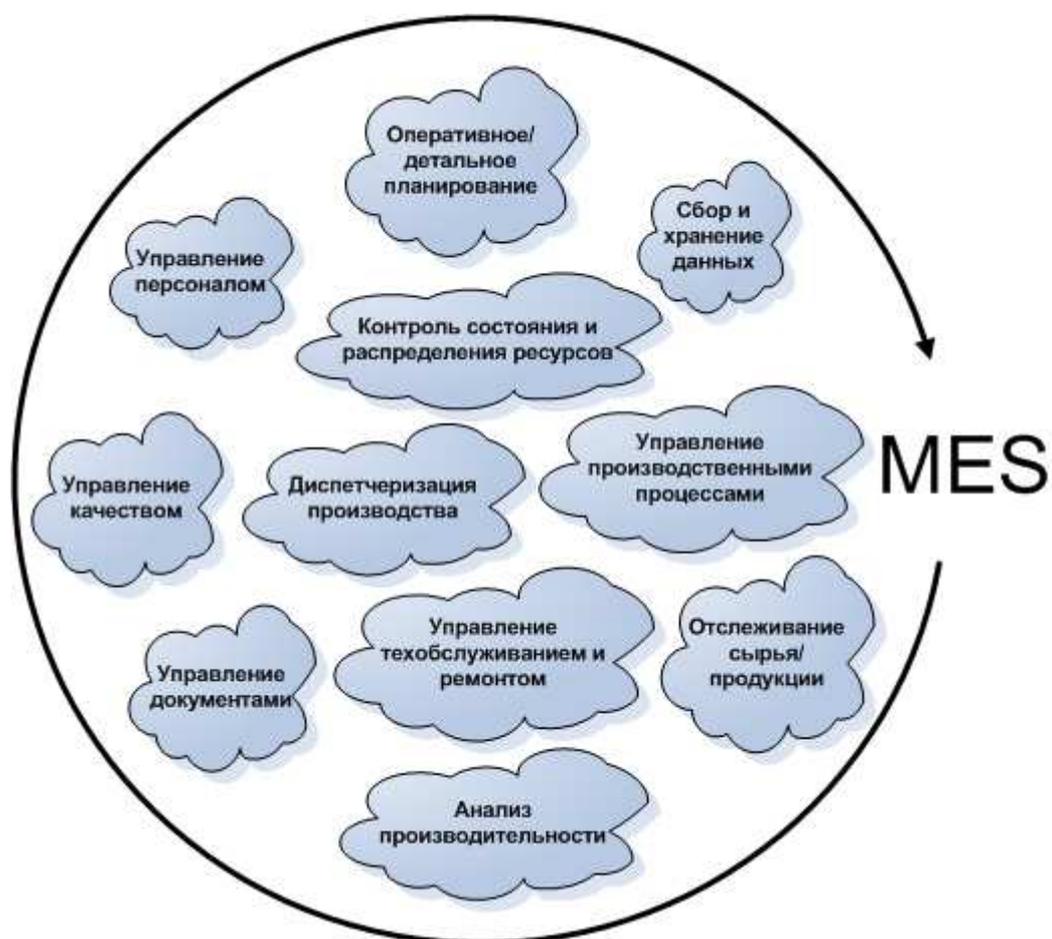


Рисунок 4 – Задачи MES-систем.

MES-модель позволяет представить производственный процесс как совокупность взаимосвязанных энергетических и материальных потоков. В процессе производства материалы и энергия перемещаются по технологическим цепочкам, при этом происходит превращение одних материалов в другие, а также возникают потери, отходы и брак. Операционный персонал производственных участков и цехов в реальном времени следит за материальными потоками, потерями производства, анализирует их причины и предпринимает меры, направленные на снижение потерь и брака. Данные о материальных потоках поступают в MES-систему от SCADA уровня, из реляционных СУБД, либо вводятся в ручную.

Популярные MES-системы: T-FACTORY и ФОБОС.

2 SCADA-системы

SCADA (Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) – 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 АСУТП, АСКУЭ, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д. Структура подобных систем показана на рисунке 5.



Рисунок 5 Структура SCADA-системы.

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.

2.1 Характеристики SCADA-систем

SCADA-системы закрывают уровень автоматизированных систем, связанный, прежде всего, с получением и визуализацией информации от программируемых логических контроллеров, распределенных систем сбора

данных и управления. Некоторые фирмы-разработчики SCADA успешно решают вопросы интеграции на уровень АСУП – MES и ERP.

Как правило, SCADA состоит из двух частей:

1. среды разработки, где инженер проектирует технологические мнемосхемы;
2. среды исполнения, необходимой для выполнения сконфигурированных мнемосхем в режиме реального времени.

Основные функции SCADA-систем:

- сбор первичной информации от датчиков и выдача управляющих воздействий через исполнительные механизмы нижнего уровня в реальном времени посредством драйверов устройств и OPC-серверов;
- обработка собранной информации в режиме реального времени;
- горизонтально-сетевое взаимодействие между диспетчерскими рабочими станциями;
- обеспечение взаимодействия с внешними приложениями (СУБД, электронные таблицы и т.д.). В информационной системе предприятия такими приложениями чаще всего являются приложения уровня MES и ERP.

Функциональные возможности SCADA-системы в основном определяются стоимостью. Большинство производителей выпускают линейку базовых продуктов с возможностью расширения за счёт приобретения дополнительных модулей. Именно они влияют на характеристики SCADA-системы. Рассмотрим некоторые из них.

Аппаратно-программная платформа. Перечень аппаратно-программных платформ, которые поддерживаются SCADA-системой. Позволяет оценить возможность распространения SCADA-системы на существующих вычислительных средствах.

Важной характеристикой SCADA-систем является количество поддерживаемых аппаратных средств: разнообразных промышленных контроллеров, модулей сбора данных и управления.

Сетевая поддержка. Современная SCADA-система должна обеспечивать возможность взаимодействия с внешними приложениями и устройства с помощью сети. Необходимо, чтобы система могла работать в стандартных сетевых средах (Arcnet, Ethernet и т.д.) с использованием общепринятых протоколов (NetBIOS, TCP/IP и др.), а также поддерживала наиболее популярные стандарты промышленных сетей (PROFIBUS, CANBUS, LON, MODBUS и т.д.).

Поддерживаемые базы данных. Практически все SCADA-системы применяют синтаксис ANSI SQL, который не зависит от типа базы данных. Это позволяет использовать практически любую базу данных.

Графические возможности. Именно от них зависит, насколько просто разработчику систем автоматизации создавать графический пользовательский интерфейс, и насколько просто им будет удобно пользоваться непосредственно оператору технологического процесса. Набор графических возможностей в различных SCADA-систем очень похож. В каждой из них есть графический объектно-ориентированный редактор со стандартным набором анимационных функций. Векторная графика позволяет осуществлять большинство графических операций над выбранным объектом, а также быстро обновлять изображение на экране.

Открытость систем. Программная система является открытой, если для нее определены и описаны используемые форматы данных и процедурный интерфейс, что позволяет подключить к ней внешние, независимо разработанные компоненты и модули. Перед фирмами-разработчиками систем управления часто встает вопрос о создании собственных (не предусмотренных в рамках систем SCADA) программных модулей и включении их в создаваемую систему. Поэтому вопрос об открытости системы является важной характеристикой SCADA-систем. Фактически открытость системы означает доступность спецификаций системных вызовов на уровне SCADA,

реализующих тот или иной системный сервис. Это может быть доступ и к графическим функциям, и функциям работы с базами данных и т.д.

Средства ввода-вывода. Современные SCADA не ограничивают выбора аппаратуры нижнего уровня, так как предоставляют большой набор драйверов или серверов ввода-вывода и имеют хорошо развитые средства создания собственных программных модулей или драйверов новых устройств нижнего уровня. Сами драйверы разрабатываются с использованием стандартных языков программирования.

Реализация задач реального времени. В большинстве случаев SCADA-системы реализованы для операционной системы MS Windows.

Наличие и качество поддержки. Необходимо обращать внимание не только на наличие технической поддержки SCADA-систем, как таковой, но и на ее качество.

Локализация. Любая система управления, имеющая интерфейс с оператором, должна допускать возможность общения с человеком на его родном языке.

Интеграция многоуровневых систем автоматизации. SCADA-системы ответственны за получение информации с нижнего уровня управления, но при этом получают информацию с верхних уровней информационной системы управления производством. Поэтому важной характеристикой любой SCADA-системы является её интеграция с приложениями верхних уровней.

2.2 Основные компоненты SCADA-систем

SCADA-система обычно содержит следующие подсистемы:

Драйверы или серверы ввода-вывода – программы, обеспечивающие связь SCADA с промышленными контроллерами, счётчиками, датчиками и другими устройствами ввода-вывода информации.

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

- стандартные протоколы динамического обмена данными (DDE);
- открытый механизм взаимодействия с базами данных ODBC;
- собственные протоколы фирм-производителей SCADA-систем, реально обеспечивающие самый скоростной обмен данными;
- OPC (OLE for Process Control) – протокол, который является стандартным и поддерживается большинством SCADA-систем [11].

Недостатками протоколов DDE являются:

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

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

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

Преимущество использования OPC:

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

Система реального времени – программа, обеспечивающая обработку данных в пределах заданного временного цикла с учетом приоритетов.

Человеко-машинный интерфейс (HMI. Human Machine Interface) — инструмент, который представляет данные о процессе в удобном для восприятия человеком виде, это позволяет оператору управлять технологическим процессом.

Программа-редактор – инструмент для разработки человеко-машинного интерфейса.

Система логического управления – модуль, обеспечивающий исполнение пользовательских программ (скриптов) логического управления в SCADA-системе.

База данных реального времени – программа, обеспечивающая сохранение истории процесса в режиме реального времени.

Система управления тревогами – программа, обеспечивающая автоматический контроль технологических событий, отнесение их к категории нормальных, предупреждающих или аварийных, а также обработку событий оператором или компьютером.

Генератор отчетов – программа, обеспечивающая создание пользовательских отчетов о технологических событиях.

Внешние интерфейсы – стандартные интерфейсы обмена данными между SCADA и другими приложениями.

В целом все популярные SCADA-системы содержат данные компоненты.

2.3 Языки программирования SCADA-систем

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

- использование языков высокого уровня (Basic-подобные языки);
- использование языков международного стандарта МЭК 6-1131/3.

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

Применение визуальных языков программирования позволяет экономить время разработки и отладки управляющих программ в разы, резко сокращает набор необходимых знаний в области классического программирования, позволяя концентрировать усилия по освоению знаний в самой прикладной области.

В международный стандарт МЭК 6-1131/3 включены 5 языков программирования:

1. FBD (Function Block Diagram) – язык функциональных блоковых диаграмм;

2. LD (Ladder Diagram) – язык релейных или релейно-контактных схем;

3. SFC (Sequential Function Chart) – язык последовательных функциональных схем;

4. ST (Structured Text) – текстовый язык программирования высокого уровня;

5. IL (Instruction List) – машиноориентированный текстовый язык программирования.

Данный стандарт разрабатывается с 1993 года Международной Электротехнической Комиссией (International Electrotechnical Commission) и давно признан ведущими производителями средств управления и автоматизации. Из этих языков три визуальных языка (FBD, SFC, LD), ориентированных на инженеров и бизнес-аналитиков, и два текстовых языка (ST, IL), ориентированных на программистов. С помощью языков МЭК-6 одинаково комфортно программируются и контроллеры, и алгоритмы человеко-машинного интерфейса (HMI), и MES-задачи.

Языки МЭК-6 сочетают в себе достаточную функциональность, простоту и предохраняют пользователя SCADA от большинства ошибок, которые нередко возникают при использовании обычных языков программирования. Реализация языков программирования в современных SCADA-системах не только полностью удовлетворяет требованиям стандарта МЭК 6-1131/3, но и

предоставляет пользователю дополнительные сервисы в виде расширенного набора библиотек функциональных блоков, реализующих типовые алгоритмы управления.

Каждая программа обладает набором аргументов, исходные данные передаются в программу через входные аргументы, а результаты вычислений возвращаются в выходных аргументах. Аргументы связываются с атрибутами каналов, т.е. с реальными входами и выходами контроллеров и модулей ввода-вывода, ячейками баз данных, либо с внутренними переменными.

Программирование и отладка программ на языках МЭК-6 производится в интегрированной среде разработки, включающей в себя несколько различных редакторов. Программы на языках FBD, LD и SFC создаются и отлаживаются в специальных визуальных редакторах, а ST и IL представляют собой более традиционные языки, программирование на которых осуществляется в текстовом редакторе. Несмотря на различия, программы на разных языках стандарта МЭК-6 могут взаимодействовать между собой. Например, программа на FBD может вызывать функциональный блок, написанный на языке ST, а внутри этого блока может вызываться подпрограмма на языке LD и т.д. Такая гибкость в выборе средств описания алгоритмов позволяет эффективно работать над одной задачей и программисту, и технологу, и инженеру-наладчику, и бизнес-консультанту, когда каждый из них выполняет свою часть работы удобным ему способом [14].

2.3.1 Язык FBD

FBD (англ. Function Block Diagram) – графический язык программирования стандарта МЭК 61131-3. Предназначен для разработки приложений программируемых логических контроллеров (ПЛК). Программа образуется из списка цепей, выполняемых последовательно сверху вниз. Цепи могут иметь метки. Инструкция перехода на метку позволяет изменять последовательность выполнения цепей для программирования условий и циклов.

При программировании используются наборы библиотечных блоков и собственные блоки, также написанные на FBD или других языках МЭК 61131-3. Блок (элемент) – это подпрограмма, функция или функциональный блок (И, ИЛИ, НЕ, триггеры, таймеры, счётчики, блоки обработки аналогового сигнала, математические операции и др.).

Каждая отдельная цепь представляет собой выражение, составленное графически из отдельных элементов. К выходу блока подключается следующий блок, образуя цепь. Внутри цепи блоки выполняются строго в порядке их соединения. Результат вычисления цепи записывается во внутреннюю переменную либо подается на выход ПЛК.

При необходимости управления вызовом блоков в них добавляются специальные входы EN (enable) и выходы ENO. Логический ноль на входе EN запрещает вызов блока. Выход ENO используется для индикации ошибки в блоке и позволяет прекратить вычисление остатка цепи.

Язык FBD прост в изучении, нагляден и удобен для прикладных специалистов, не имеющих специальной подготовки в области информатики. Жесткая последовательность выполнения приводит к простой внутренней структуре команд, которая транслируется в быстрый и надежный код.

Каждый FBD блок имеет графическое изображение, принятое при разработке функциональных схем электронных устройств. Представлены они в виде прямоугольников и являются собой единичной операцией над входными переменными.

Выходные контакты могут подключаться к выходным и внутренним переменным. Кроме того, можно передавать выходной сигнал на вход другому функциональному блоку или же функции.

Переменные и входы соединяют специальными линиями связи. Эта линия и указывает распространение сигнала FBD программы.

Идеология программирования на языке FBD подразумевает, что время исполнения каждой программы должно быть вполне определенным, т.е.

детерминированным. Другими словами ни одна программа не имеет права заикнуться на неопределенное время, например на ожидании какого-либо события.

Данный язык программирования, помимо прочего, может использоваться для описания шагов и переходов в языке SFC. Функциональные блоки объединяют данные и методы, чем подобны объектно-ориентированным языкам программирования, но они не поддерживают наследование и полиморфизм.

Описание FBD блоков в целом схоже с описанием функций и методов в привычных языках программирования. К типовым FBD блокам относятся блок таймера, ПИД-регулятора, блок секвенсора, триггера, генератора импульсов, фильтра, и т. п.

FBD программа формируется в специальном LCADA-конфигураторе и загружается на ПЛК[20]. Пример программы на FBD, показан на рисунке 6.

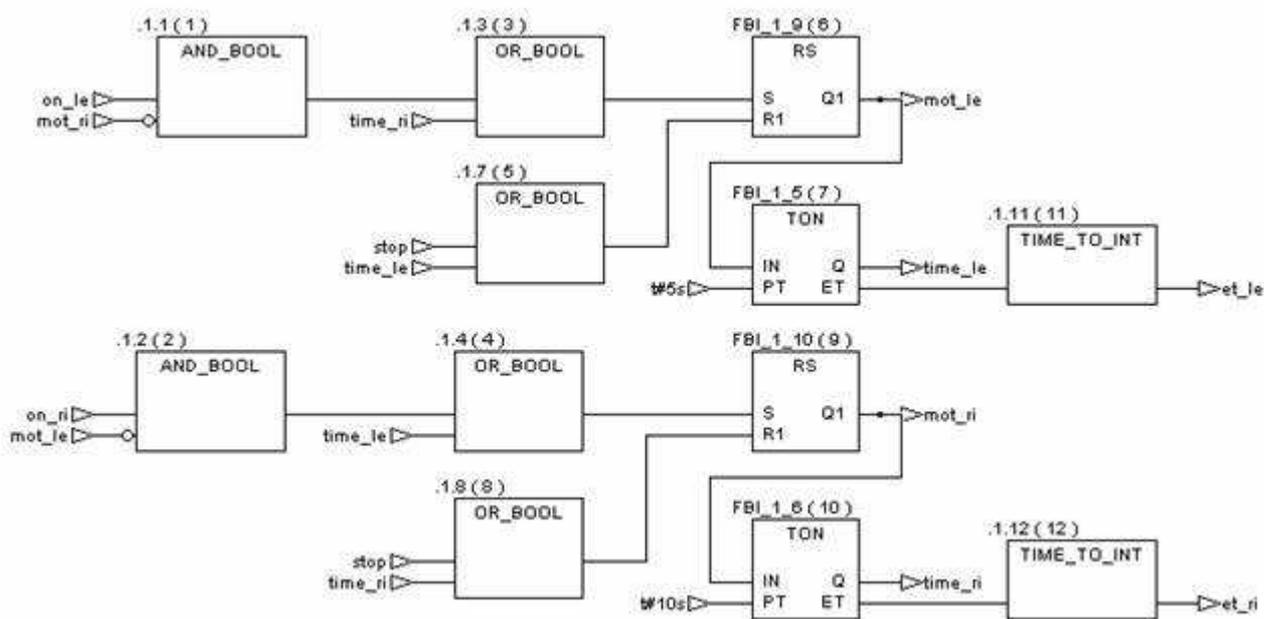


Рисунок 6 – Пример программы на FBD языке.

Существует много практических реализаций языка FBD с определенными расширениями или ограничениями.

Одним из вариантов FBD является язык программирования CFC (Continuous Function Chart). Он позволяет произвольно задавать порядок выполнения блоков. Диаграммы CFC дают программисту большую свободу действий.

2.3.2 Язык LD

LD (англ. Ladder Diagram) — графический язык, реализующий структуры электрических цепей. Лучше всего LD подходит для построения логических переключателей, но достаточно легко можно создавать и сложные цепи – как и в FBD. Кроме того, LD достаточно удобен для управления другими компонентами программ.

Диаграмма LD состоит из ряда цепей. Пример LD диаграммы приведён на рисунке 7.

Слева и справа схема ограничена вертикальными линиями — шинами питания. Между ними расположены цепи, образованные контактами и обмотками реле, по аналогии с обычными электронными цепями. Слева любая цепь начинается набором контактов, которые посылают слева направо состояние «ON» или «OFF», соответствующие логическим значениям ИСТИНА или ЛОЖЬ. Каждому контакту соответствует логическая переменная. Если переменная имеет значение ИСТИНА, то состояние передается через контакт. Иначе правое соединение получает значение выключено («OFF»).

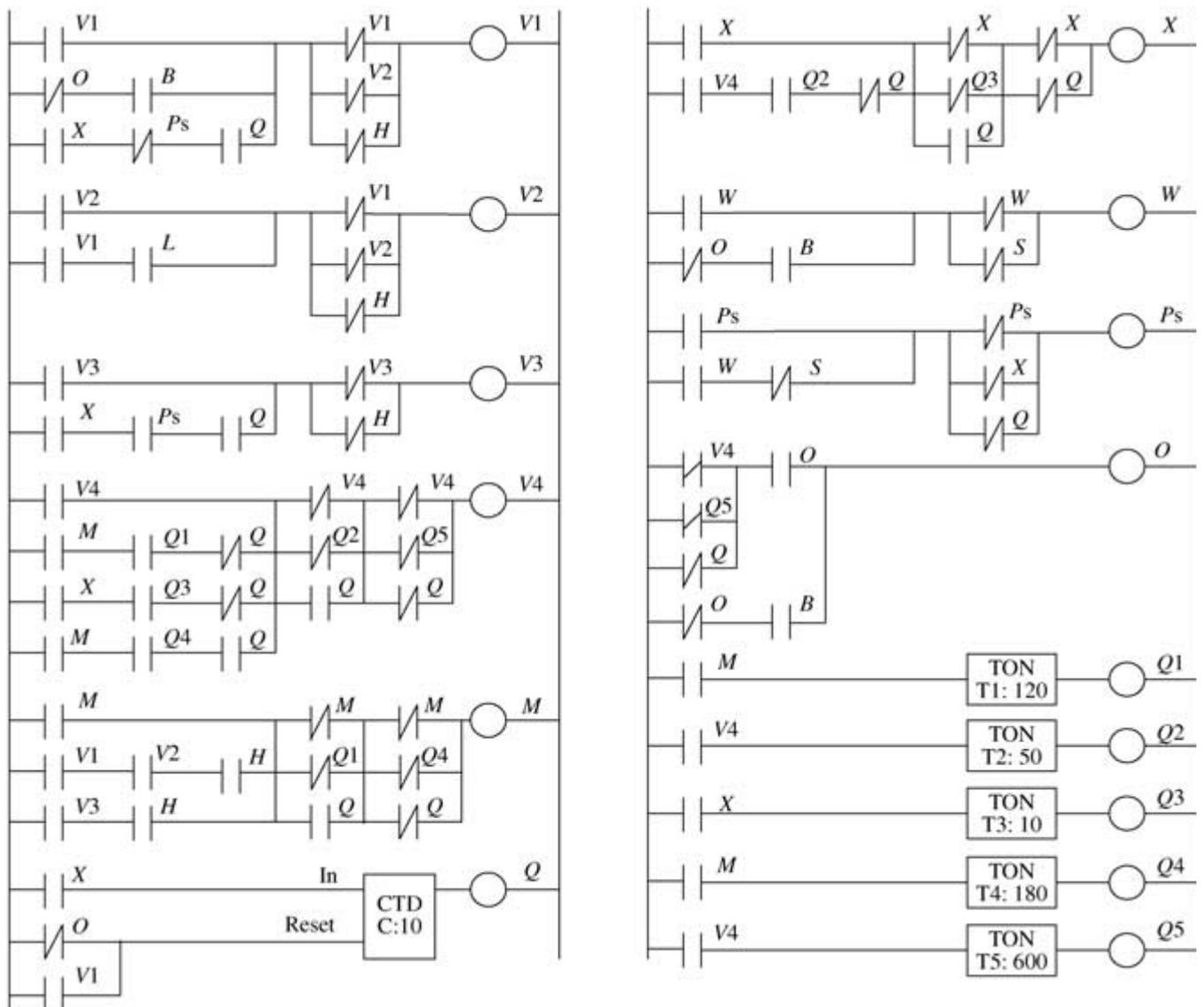


Рисунок 7 – Пример LD диаграммы.

Контакт. Контакты обозначаются двумя параллельными линиями и могут иметь состояния «ON» или «OFF». Эти состояния соответствуют значениям ИСТИНА или ЛОЖЬ. Каждому контакту соответствует логическая переменная. Если значение переменной ИСТИНА, то контакт замкнут. Контакты могут быть соединены параллельно, тогда соединение передает состояние «ON», когда хотя бы одна из ветвей передает «ON». Если контакты соединены последовательно, то для того, чтобы соединение передало «ON», необходимо, чтобы оба контакта передавали «ON». Это соответствует электрической параллельной и последовательной схеме. Контакт может быть инвертируемым.

Такой контакт обозначается с помощью символа $|/|$ и передает состояние «ON», если значение переменной ЛОЖЬ.

Обмотка. В правой части схемы может находиться любое количество обмоток (реле), которые обозначаются круглыми скобками (). Они могут соединяться только параллельно. Обмотка передает значение соединения слева направо и копирует его в соответствующую логическую переменную. В целом цепь может быть либо замкнутой (ON), либо разомкнутой (OFF). Это как раз и отражается на обмотке и соответственно на логической переменной обмотки (ИСТИНА/ЛОЖЬ). Обмотки также могут быть инверсными, если обмотка инверсная (обозначается символом (/), тогда в соответствующую логическую переменную копируется инверсное значение.

Функциональные блоки в LD. Кроме контактов и обмоток, в LD можно использовать функциональные блоки и программы. Они должны иметь логические вход и выход и могут использоваться так же, как контакты.

SET и RESET обмотка. Обмотки могут быть с «самофиксацией» типов SET и RESET. Обмотки типа SET обозначаются буквой «S» внутри круглых скобок (S). Если соответствующая этой обмотке переменная принимает значение ИСТИНА, то она навсегда (до сброса R) сохраняет его. Обмотки типа RESET обозначаются буквой R. Если соответствующая переменная принимает значение ЛОЖЬ, то она навсегда (до установки S) сохраняет его.

LD в качестве FBD. При работе с LD можно с помощью контакта управлять другими программами. Во-первых, можно использовать обмотку для передачи значения глобальной переменной, которая будет использоваться в другом месте. Кроме того, можно вставить вызов прямо в схему LD. Такой блок может быть оператором, функцией, программой или функциональным блоком, который имеет добавочный вход, обозначаемый EN. Вход EN всегда логического типа, и программа выполняется, только когда значение EN=ИСТИНА. Программа встраивается в схему параллельно обмоткам, и вход

EN соединяется ответвлением. Использование таких блоков делает LD схему похожей на FBD схему [12].

2.3.3 Язык SFC

SFC — графический язык, предназначенный для написания программ последовательного управления технологическим процессом, описывающий его в форме близкой к диаграмме состояний. Для этого действия связываются с шагами (этапами), а последовательность работы определяется условиями переходов между шагами [16]. Пример программы на языке SFC приведён на рисунке 8.

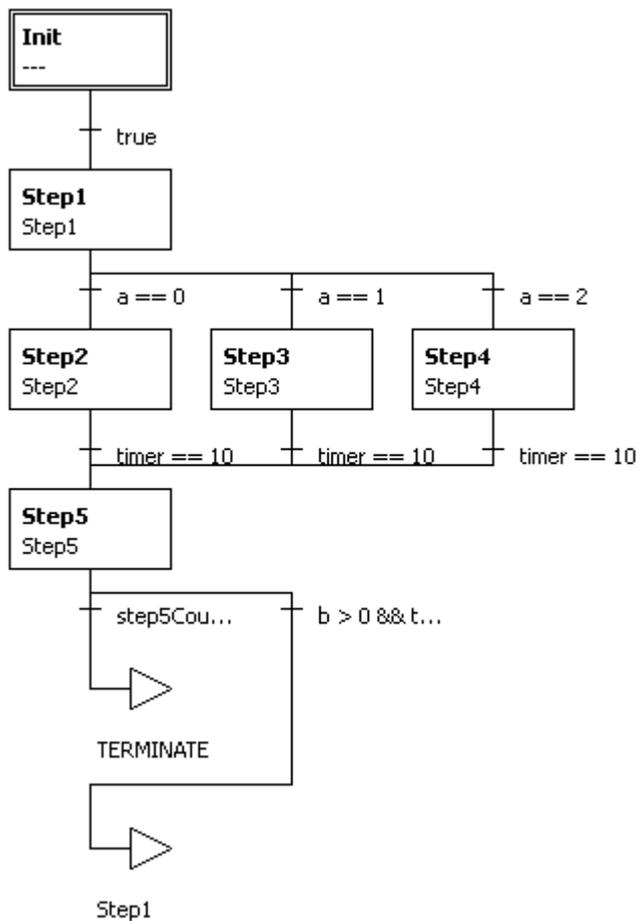


Рисунок 8 – Пример программы на языке SFC.

Основными элементами языка являются:

– состояния, в которых выполняются определенные действия, одновременно могут быть активны несколько состояний, одно из состояний является начальным;

– переходы из состояния в состояние, для каждого перехода задаются логическое условие перехода к следующему шагу;

– альтернативное ветвление алгоритма, когда из текущего состояния возможны переходы к нескольким состояниям, при этом каждому переходу соответствует своё логическое условие и при выполнении алгоритма производится только один из альтернативных переходов;

– параллельное ветвление, в отличие от альтернативного имеет общее условие перехода на несколько параллельно работающих веток;

– переход к заданному состоянию;

– остановка процесса.

Шаг. Программа на языке SFC состоит из набора шагов, связанных переходами. Существуют 2 вида шагов:

– Шаг простого типа (упрощенный SFC) может включать единственное действие. Графический флажок (небольшой треугольник в верхнем углу шага) показывает, пустой шаг или нет.

– МЭК шаг (стандартный SFC) связан с произвольным числом действий или логических переменных. Связанные действия располагаются с правой стороны от шага.

Действие. Действие может содержать список инструкций на IL или ST, схемы на FBD или LD, или снова схемы на SFC. При использовании простых шагов действие всегда связывается с этим шагом. Для того чтобы редактировать действие, необходимо дважды щелкнуть левой клавишей мышки на шаге. Помимо основного действия, шаг может включать одно входное и одно выходное действие. Действия МЭК шагов показаны в Организаторе Объектов, непосредственно под вызывающей их ROU. Редактирование действия запускается двойным щелчком мыши или клавишей Enter. Новые

действия добавляются командой главного меню Project Add Action. Одному шагу можно сопоставить до 9 действий.

Входное или выходное действие. В шаг можно добавить входное и выходное действие. Входное действие выполняется один раз при активизации шага, выходное – при деактивизации.

Шаг, который имеет входное действие, обозначается буквой «Е» в левом нижнем углу, шаг с выходными действиями – буквой «Х» в правом нижнем углу.

Входные и выходные действия могут описываться на любом языке. Для того чтобы отредактировать входное или выходное действие, надо дважды щелкнуть мышкой в соответствующем углу шага.

Переход/условие перехода. Между шагами находятся так называемые переходы. Условием перехода может быть логическая переменная или константа, логический адрес или логическое выражение, описанное на любом языке. Условие может включать серию инструкций, образующих логический результат, в виде ST выражения либо на любом другом языке. Но условие не должно содержать присваивания, вызов программ и экземпляров функциональных блоков.

В редакторе SFC условие перехода можно записать непосредственно около символа перехода либо в отдельном окне редактора для ввода условия. Условие, заданное окне редактора, предпочтительнее.

Активный шаг. После вызова программы на SFC начальный шаг (шаг, выделенный двойной рамкой) выполняется первым. Шаг, выполняемый в данный момент, называется активным. Действия, связанные с активным шагом, выполняются один раз в каждом управляющем цикле. В режиме online активные шаги выделяются синим цветом. Следующий за активным шагом шаг станет активным, только когда условие перехода к этому шагу будет истинно.

В каждом управляющем цикле будут выполнены действия, содержащиеся в активных шагах. Далее проверяются условия перехода, и, возможно, уже

другие шаги становятся активными, но выполняться они будут уже в следующем цикле.

Шаг МЭК. В отличие от упрощенного SFC МЭК шаги могут включать несколько действий. Действия МЭК шагов описываются отдельно от них и могут неоднократно использоваться в пределах данного POU, для чего их надо связать с шагом с помощью команды главного меню Extras Associate action. Кроме действий с шагом, можно связывать логические переменные.

С помощью классификаторов действия логические переменные могут активироваться и деактивироваться, возможно, с задержкой времени. Например: действие может продолжать работу, даже если запустивший его шаг утратил активность; с помощью классификатора S (установка) можно программировать параллельные процессы и т.д. Логическая переменная, связанная с шагом, получает значение ИСТИНА при каждой активации шага. Действие, связанное с МЭК шагом, описывается справа от него в блоке, состоящем из двух частей. Левая часть этого блока содержит классификатор, возможно, с константой времени, а правая часть содержит имя действия или логической переменной.

В режиме online все активные действия выделяются синим цветом, подобно активным шагам, благодаря чему достаточно легко проследить ход выполнения процесса после каждого управляющего цикла.

При выполнении шага сначала производится деактивация действий, затем выполняются активные действия в алфавитном порядке. Для того чтобы использовать шаги с МЭК действиями, необходимо установить опцию Extras «se IEC-Steps и подключить к проекту специальную библиотеку Iecsfc.lib. В Организаторе объектов действия показаны непосредственно под SFC POU's, которые их вызывают. Новые действия можно создавать с помощью команды Project Add Action.

Альтернативная ветвь. Две и более ветви SFC могут быть альтернативными. Каждая альтернативная ветвь должна начинаться и

заканчиваться переходом. Альтернативные ветви могут содержать параллельные ветви и другие альтернативные ветви. Альтернативная ветвь начинается горизонтальной линией (начало альтернативы), а заканчивается горизонтальной линией (конец альтернативы) или переходом на произвольный шаг (jump). Если шаг, который находится перед линией альтернативного начала, активен, то первые переходы альтернативных ветвей начинают оцениваться слева направо. Таким образом, первым активируется тот шаг, который следует за первым слева истинным переходом.

Параллельные ветви. Две и более ветви SFC могут быть параллельными. Каждая параллельная ветвь должна начинаться и заканчиваться шагом. Параллельные ветви могут содержать альтернативные ветви и другие параллельные ветви. Параллельная ветвь наносится двойной горизонтальной линией (параллельное начало) и заканчивается двойной горизонтальной линией (конец параллели) или переходом на произвольный шаг (jump). Если шаг активен, условие перехода после этого шага истинно и за этим переходом следуют параллельные ветви, то активируются первые шаги этих ветвей. Эти ветви выполняются параллельно друг другу. Шаг, находящийся после параллельных ветвей, становится активным только тогда, когда все предыдущие шаги активны и условие перехода истинно.

Переход на произвольный шаг (Jump). Переход на произвольный шаг – это соединение на шаг, имя которого указано под знаком «jump». Такие переходы нужны для того, чтобы избежать пересекающихся и идущих вверх соединений.

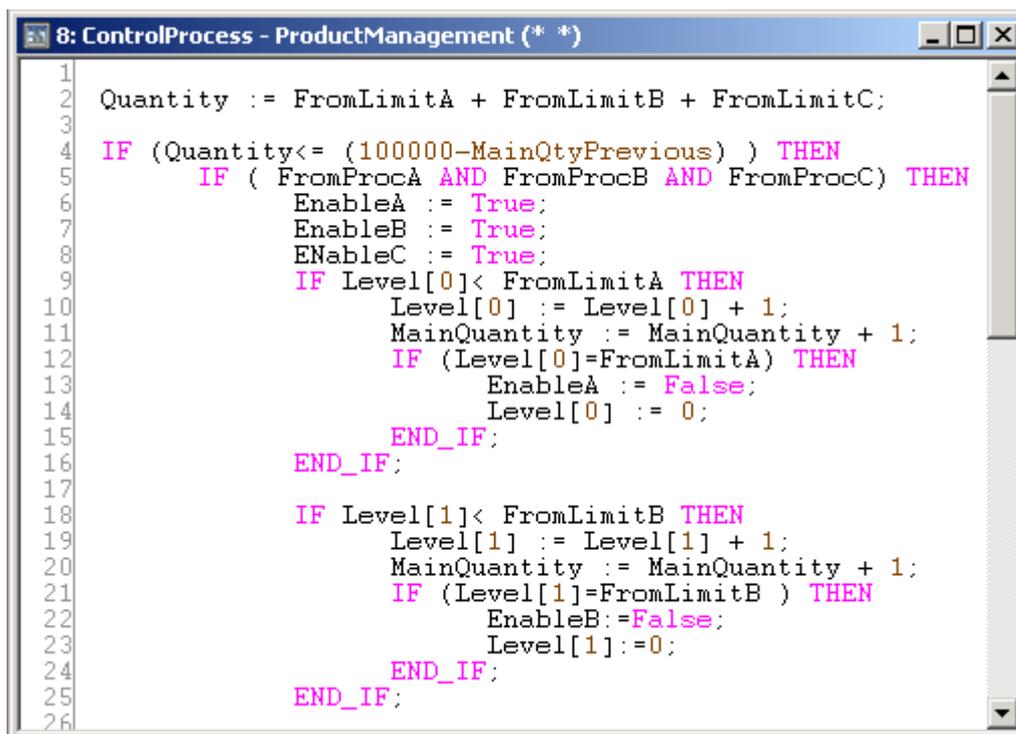
Язык SFC позволяет описывать технологические процессы, поэтому популярен при написании систем управления [16].

2.3.4 Язык ST

Стандарт МЭК включает высокоуровневый язык ST, по выразительной мощности не уступающий языкам С и Паскаль. К сожалению, в большинстве систем МЭК программирования он не реализован вовсе либо имеет массу ограничений. Как правило, это ограничение числа поддерживаемых типов

данных, упрощенный интерпретирующий транслятор или промежуточный код, ограничения на доступ к аппаратным ресурсам, невозможность обрабатывать прерывания, выполнять динамическое распределение памяти, управлять процессами и др. На самом деле, это не ограничения языка ST. Это именно ограничения среды программирования[15].

Некоторым особняком в ряду комплексов МЭК стоит CoDeSys производства 3S-Smart Software Solution GmbH. Его уникальность состоит в том, что он изначально задумывался как инструмент для профессионалов. Исходя из собственных соображений, при адаптации CoDeSys для конкретного ПЛК, изготовитель может наложить ряд ограничений на возможности пользователя. Но изначально их нет. Пример программы на языке ST, приведён на рисунке 9.



```
8: ControlProcess - ProductManagement (* *)
1
2 Quantity := FromLimitA + FromLimitB + FromLimitC;
3
4 IF (Quantity <= (100000 - MainQtyPrevious)) THEN
5     IF (FromProcA AND FromProcB AND FromProcC) THEN
6         EnableA := True;
7         EnableB := True;
8         EnableC := True;
9         IF Level[0] < FromLimitA THEN
10            Level[0] := Level[0] + 1;
11            MainQuantity := MainQuantity + 1;
12            IF (Level[0] = FromLimitA) THEN
13                EnableA := False;
14                Level[0] := 0;
15            END_IF;
16        END_IF;
17
18        IF Level[1] < FromLimitB THEN
19            Level[1] := Level[1] + 1;
20            MainQuantity := MainQuantity + 1;
21            IF (Level[1] = FromLimitB) THEN
22                EnableB := False;
23                Level[1] := 0;
24            END_IF;
25        END_IF;
26
```

Рисунок 9 – Пример программы на языке ST.

Безусловно, в CoDeSys можно использовать внешние библиотеки, реализованные на C, но необходимость в этом возникает крайне редко. Практически все, что можно написать на C, в CoDeSys можно написать на ST,

без изменения методологии проектирования. CoDeSys имеет встроенный компилятор и генераторы машинного кода для всех популярных семейств микропроцессоров. Внешний C компилятор не даст выигрыша по быстродействию, но значительно усложнит разработку [16].

Особенность стандарта МЭК состоит в том, что он включает только требование однозначности выполнения описанных в нем конструкций. Многие воспринимают это как возможность называть МЭК системой программирования среду, поддерживающую только часть стандарта. Однако справедлив и обратный подход: МЭК система может включать все, что описано в стандарте, плюс дополнительные возможности.

Типы данных. В CoDeSys реализованы все стандартные типы данных МЭК 61131-3. Все типы данных доступны во всех 6 поддерживаемых языках. Специальных расширений для ST нет.

Инициализация переменных. По умолчанию, все переменные инициализируются нулем. Иное значение переменной можно указать явно при ее объявлении.

Прямая адресация. Помимо общей памяти данных в контроллере предусмотрены 3 специальные области памяти. Это область входов, выходов и прямо-адресуемая область. Эти области служат для связи системы исполнения контроллера и пользовательской программы.

Многозадачность. Многозадачность в CoDeSys доступна даже для ПЛК, выполненных на 8 разрядных микропроцессорах без ОС. В простейшем случае, вместо вызова единственной PLC_PRG, система попеременно подставляет вызовы разных пользовательских программ. Конечно, с учетом их приоритетов и времен циклов выполнения. Так реализуется не вытесняющая многозадачность. В многозадачных проектах необходимо сопоставить программы задачам с помощью менеджера задач в CoDeSys. Если этого не сделать, то проект будет однозадачным, с единственной главной функцией PLC_PRG.

Библиотеки. CoDeSys поддерживает 2 вида библиотек.

Первый вид – это внутренние библиотеки. Любой проект можно сохранить как библиотеку. Внутренние библиотеки пишутся и хранятся на языках МЭК. Они компилируются вместе с проектом. Вы можете запретить редактирование и просмотр текстов внутренних библиотек пользователем. Специальный менеджер библиотек позволяет защитить вашу библиотеку лицензией и сделать ее платной.

Второй тип – это внешние библиотеки. Их код откомпилирован внешними средствами и включен в систему исполнения (например, системные библиотеки) или размещен в obj файлы.

Внешние библиотеки могут иметь еще и ‘внутреннюю’ реализацию (на языках МЭК). Это позволяет написать заглушки системных функций, работающие в режиме эмуляции контроллера в среде программирования CoDeSys.

Обработка ошибок. CoDeSys имеет специальный набор функции с фиксированными именами, позволяющими организовать обработку критических ошибок в пользовательской программе.

По числу типов данных ST превосходит C. Редактор ST в CoDeSys поддерживает синтаксическое цветовое выделение и имеет средства быстрого ввода[15].

2.3.5 Язык IL

IL – язык программирования стандарта IEC61131-3. Предназначен для программирования промышленных контроллеров. По синтаксису напоминает ассемблер.

Программы, написанные на языке IL, это последовательности высказываний. Высказывания состоят из операторов и операндов, которые могут быть переменными, константами или метками. Кроме того, программы могут дополняться комментариями. Пример программы на языке IL показан на рисунке 10.

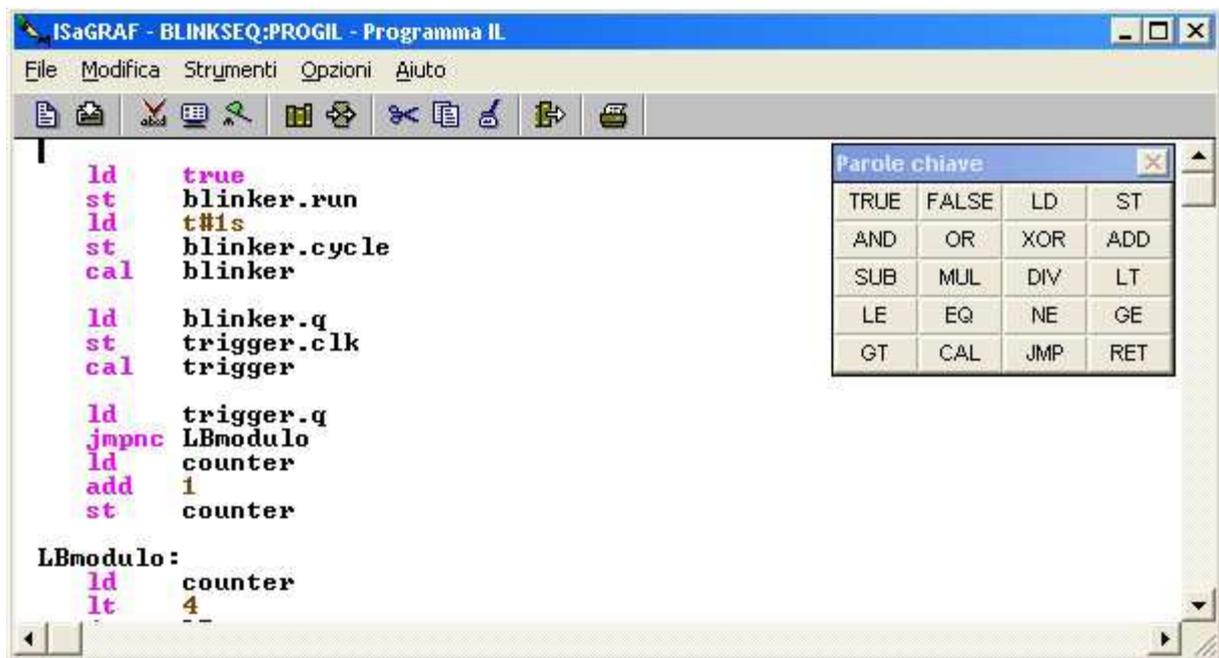


Рисунок 10 – Пример программы на языке IL.

Некоторые операторы могут быть дополнены модификаторами. Существует два модификатора: N и C. Модификатор N изменяет операцию на такую же с отрицательным аргументом, т.е. отрицательным значением операнда. Операторы, дополненные модификатором C, исполняются только при условии, что значение CR истинно.

Использование скобок позволяет заставить подвыражения вычисляться первыми, следовательно, избежать вспомогательные переменные или дополнительные операции загрузки/сохранения [3].

Типизирование данных является общим элементом стандарта с целью предотвращения ошибок на ранней стадии разработки.

Язык IL имеет все недостатки, которые присущи другим низкоуровневым языкам программирования: сложность и высокую трудоемкость программирования, трудность модификации написанных на нем программ, малую степень «видимого» соответствия исходного текста программы и решаемой задачи.

Ввиду своей ненаглядности, IL практически не используется для программирования комплексных алгоритмов автоматизированного управления,

но часто применяется для кодирования отдельных функциональных блоков, из которых впоследствии складываются схемы FBD или SFC. При этом IL позволяет достичь высокой оптимальности кода: программные блоки, написанные на IL, имеют высокую скорость исполнения и наименее требовательны к ресурсам контроллера.

3 Обзор SCADA-систем, представленных на рынке

На мировом рынке представлено большое количество различных SCADA-систем. Они отличаются друг от друга по области применения, по функциональному составу и информационной мощности. В таблице 1 приведены наиболее популярные SCADA-системы применяемые при проектирование систем автоматизации на мировом и российском рынке.

Таблица 1 – Популярные SCADA-системы.

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

Рассмотрим более подробно некоторые из SCADA-системы, представленные в таблице.

3.1 SCADA-система InTouch

SCADA-система компании Wonderware уже более 25 лет занимает лидирующие позиции среди человеко-машинных интерфейсов.

Программное обеспечение представляет собой открытую и расширяемую управляющую систему, которая обеспечивает быстрое создание стандартизованных, многократно используемых приложений для графической визуализации и их централизованное развертывание в рамках автоматизации всего предприятия.

SCADA-система InTouch используется на более чем одной трети, промышленных объектов в мире, обеспечивает хорошую эффективность при решении ключевых задач:

- простота проектирования;
- операционная гибкость;

- управление в режиме реального времени;
- повышение гибкости;
- снижение затрат, рисков и повышение безопасности.

Подход компании Wonderware к представлению информации позволяет снизить время обработки проблемы и уделить больше внимания анализу ситуации в целом и данным. Это достигается за счёт наличия удобных информационных дисплеев и усовершенствованной системы аварийных тревог, позволяющей сконцентрироваться на реальной проблеме. Правильное использование цветовой гаммы позволяет повысить эффективность работы оператора, уменьшить влияния отвлекающих факторов, фактора усталости.

Пример приложения для SCADA InTouch показан на рисунке 11.

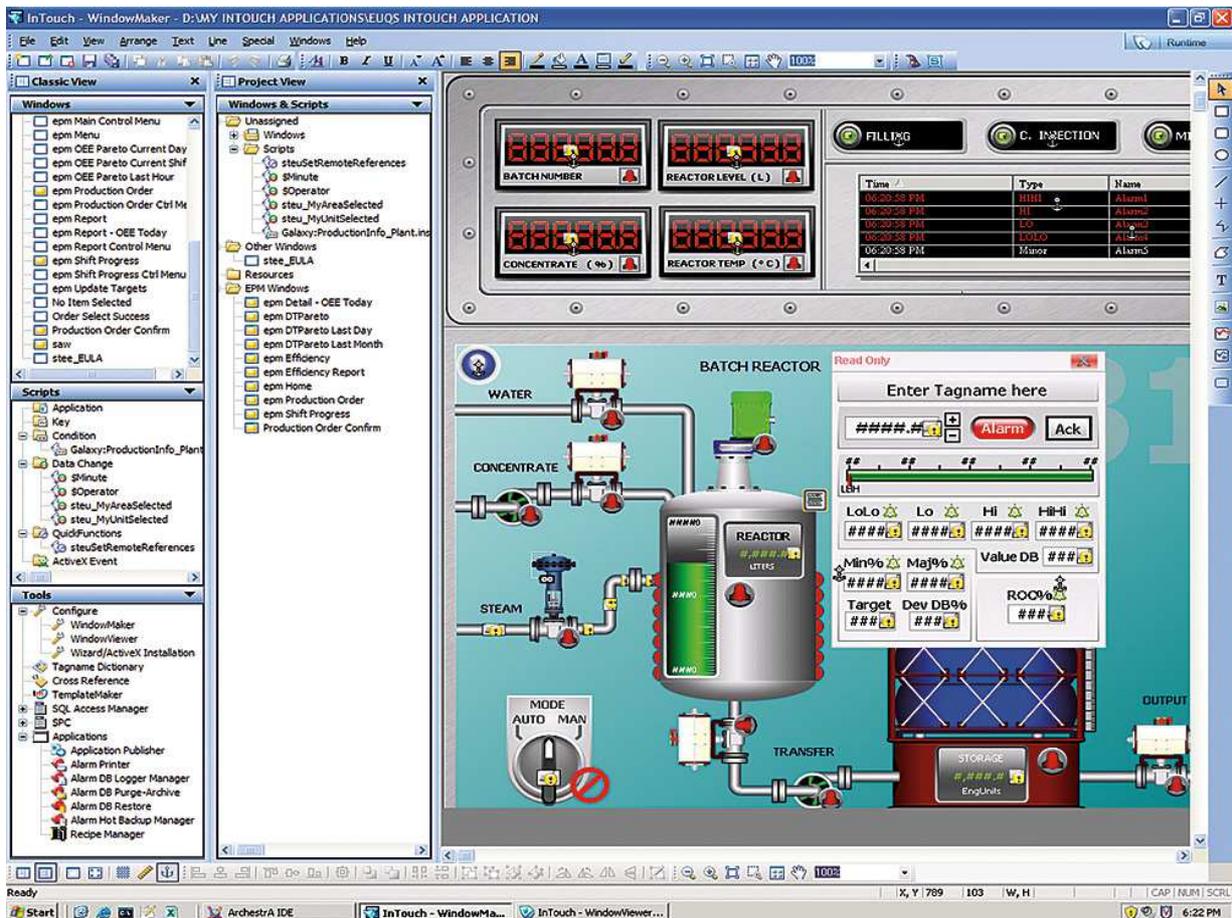


Рисунок 11 – Приложение для SCADA InTouch.

SCADA InTouch позволяет существенно повысить квалификацию операторов путем улучшения навыков по выявлению тенденций на стадии их формирования, выработке логически взвешенных предположений, внесению корректив для своевременного решения проблем и предупреждения чрезвычайных ситуаций.

Особенности и преимущества InTouch:

Усиленный контроль процессов.

Стили элементов (Element Styles). Как правило, на разработку приложений требуется много времени и усилия многих инженеров. Данное обстоятельство может привести к нарушению единообразия применяемых стандартов, цветовых гамм, текста и индикаторов аварийных сигналов или событий. Стили элементов позволяют создать клиентоориентированные приложения на предприятии, независимо от того, когда и кем он был разработан.

Высокая производительность проектных отделов. Передовые инструменты проектирования позволяют ускорить разработку приложений, а также поиск и устранения неполадок, повысить производительность и упростить обслуживание приложений.

Мастера символов (Symbol Wizards). Новый запатентованный функционал Symbol Wizards обеспечивает максимальную гибкость и стандартизацию в процессе конфигурирования символов с одновременным снижением трудозатрат на разработку и обслуживание приложений.

Контентоориентированный редактор скриптов. Режим работы редактора сценариев с функцией «Автоматического завершения» позволяет предугадать следующий шаг инженера и вырабатывает интеллектуальные предложения завершения написания скриптовой функции, наиболее совпадающие с намерениями специалиста. InTouch 2014 объединяет всю работу по созданию сценариев в рамках единой среды разработки и помогает сократить время на разработку сценариев на 40 %.

Поддержка новых инновационных технологий. Обеспечивая поддержку самых современных технологий, программное обеспечение InTouch поддерживает HTML5, глобальный стандарт языка разметки для структурирования и представления контента в веб-браузере. InTouch Access Anywhere позволяет пользователям надежно устранять неполадки в оборудовании предприятия в любом месте, на любом устройстве и в любое время. Мобильные работники могут загрузить и использовать приложения InTouch в любой точке предприятия, что позволяет им получать всю необходимую информацию и следить за всеми важными процессами[8].

Достоинством SCADA-системы Intouch является:

- высокая скорость работы благодаря механизму, который динамически регулирует скорость опроса входных сигналов (опрос происходит только при изменении значения контролируемого параметра);
- архитектура клиент-сервер для эффективной работы в сети с БД;
- открытость – возможность добавлять и использовать компоненты других фирм вследствие поддержки технологий ActiveX и OPC;
- интеграция с другими программными пакетами фирмы Wonderware и простой обмен данными с программными пакетами Microsoft Office, Microsoft Visual Basic и др.;
- большое число серверов ввода/вывода – более 600;
- возможность создания собственных библиотек алгоритмов;
- автоматический контроль качества сигналов с датчиков и контроллеров;
- распределённая система обнаружения и регистрации аварийных ситуаций одновременно поддерживает множество серверов (провайдеров) аварийных ситуаций, что даёт возможность операторам видеть информацию об авариях во многих удалённых местах синхронно [11].

Всё это делает SCADA-систему InTouch лидеров в области автоматизации производства в мире.

3.2 SCADA-система Genesis32

SCADA-система Genesis32 обладает традиционным набором свойств и характеристик SCADA-систем, а также содержит большой перечень новых, появившихся недавно программных компонентов сквозной автоматизации производства. Пример экрана в SCADA Genesis32 показан на рисунке 12.

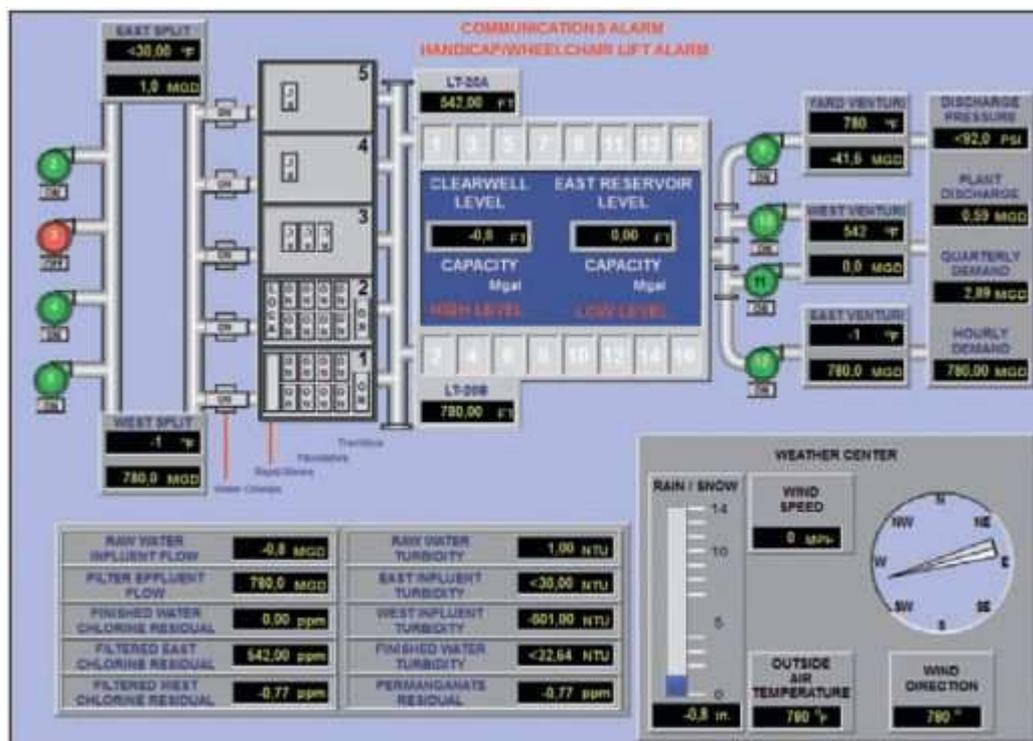


Рисунок 12 – Пример экрана в SCADA Genesis32.

Функциональные возможности:

- автоматизированная разработка программного обеспечения системы автоматизации без навыков программирования;
- средства сбора первичной информации от устройств нижнего уровня;
- средства управления и регистрации сигналов об аварийных ситуациях;
- средства хранения информации с возможностью ее последующей обработки (реализуется через интерфейсы к наиболее популярным базам данных);
- средства обработки первичной информации;

– средства визуализации представления информации в виде графиков и гистограмм;

– возможность работы прикладной системы с наборами параметров, рассматриваемых как единое целое [19].

SCADA-система GENESIS32 реализована на платформе MS Windows. Именно такие системы предлагают наиболее полные и легконаращиваемые человеко-машинные интерфейсные средства. Сетевая поддержка системы реализована с помощью технологии GenBroker, которая создана для построения устойчивых сетевых соединений и, благодаря использованию протоколов TCP/IP и SOAP/XML, обеспечивает возможность удалённого взаимодействия.

Встроенные командные языки и дополнительные средства обработки.

Система может работать с элементами управления ActiveX и объектами OLE. Встроена поддержка редакторов сценарных процедур Microsoft Visual Basic for Applications, VBScript и JScript и возможность работать с приложениями MS Office. Также в систему включены:

– инструменты отладки и управления проектами, позволяющие осуществлять управление удаленными приложениями со станции разработчика;

– конвертеры из популярных мировых SCADA-систем;

– средства работы с промышленными сетями.

Поддерживаемые базы данных. SCADA-система GENESIS32 использует синтаксис ANSI SQL, который не зависит от типа базы данных. Таким образом реализована поддержка работы с мировыми производителями СУБД.

В новой версии GENESIS32 V9.1 интегрированы мощные технологии, направленные на совместимость данных уровня управления предприятия (SAP), а также контроля и взаимодействия с инфраструктурой ИТ (поддержка протокола SNMP).

Компания ICONICS постоянно совершенствует свою продукцию. Используя новые технологии в области инструментальных средств комплексной автоматизации и управления интеллектуальными предприятиями,

она достигла небывалой популярности. Также компания выпустила на рынок новые полнофункциональные решения для 64-битной платформы Windows.

3.3 Программный продукт TIA Portal

Результатом многолетних усилий и значительных инвестиций, вложенных в разработку, стал новый программный продукт компании Siemens, получивший название Totally Integrated Automation Portal или кратко TIA Portal.

TIA Portal, в дополнение к STEP 7, включает в себя WinCC для разработки и исполнения в режиме реального времени проекта визуализации процесса, а также он-лайн справку как для WinCC, так и для STEP 7. Более подробно рассмотрим SCADA-систему WinCC.

3.3.1 SCADA-система WinCC

SCADA система SIMATIC WinCC немецкой компании Siemens является эффективной основой для построения информационной системы предприятия благодаря современным технологиям, таким как мощная интегрированная база данных для архивирования данных процесса, инструменты для обработки, передачи и анализа информации и открытые интерфейсы для интеграции. Таким образом, WinCC предлагает преимущества, которые позволят компаниям успешно конкурировать на рынке [6].

Пример экрана SCADA системы SIMATIC WinCC показан на рисунке 13.

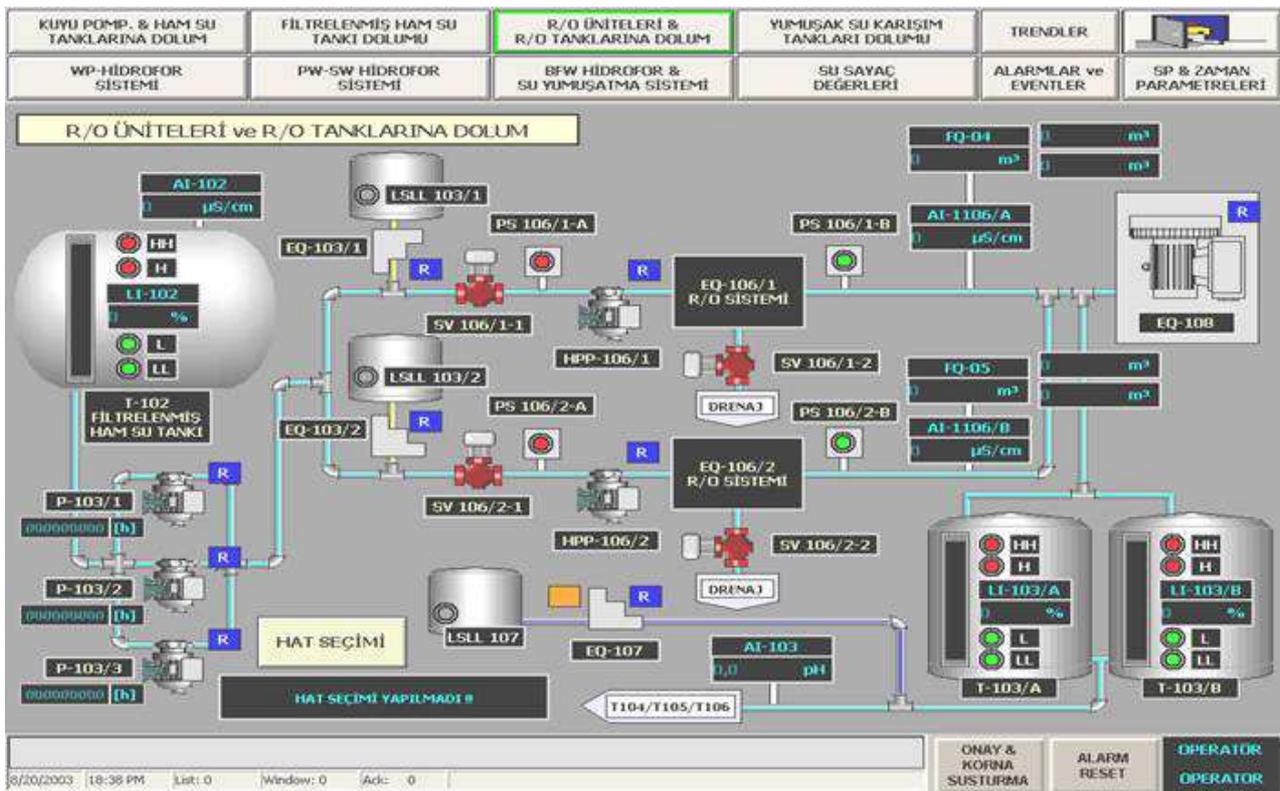


Рисунок 13 – Пример экрана SCADA система SIMATIC WinCC.

Базовые функциональные модули SIMATIC WinCC:

WinCC Explorer – быстрый обзор всех данных проекта, глобальных установок, запуска редакторов и режима Runtime, конфигурация системы «Клиент-сервер».

Graphics Designer – разработка мнемосхем с помощью стандартных элементов, ActiveX объектов, объектов из встроенной библиотеки. Динамика может задаваться с помощью прямой привязки к тегам, динамических диалогов, визардов, скриптов на языке ANSI-C или VBS.

Alarm Logging – сбор и архивация системных сообщений. WinCC поддерживает два метода генерации сообщений: с помощью тегов ПЛК и в виде пакетов сообщений (штамп времени ПЛК). Сообщения могут генерировать звуковые сигналы.

Tag Logging – сбор, архивирование и сжатие измеряемых величин. База данных основана на MS SQL Server. Архивация производится циклически или управляется событиями в системе. Может производиться архивация отдельных

тегов или целиком блоков данных ПЛК. Данные из архива могут отображаться в виде графиков или таблиц. Возможно создание долговременных архивов данных.

Report Designer – генерация отчетов в свободно программируемом формате, управляемая событиями или по времени. Возможна генерация протоколов сообщений, измеряемых величин и пользовательских отчетов. В отчет можно включать данные из CSV файлов и баз данных. Возможен предварительный просмотр отчетов и сохранение их в файл.

Global Scripts – программирование действий, производимых с графическими объектами, а также скриптов, выполняющихся в фоновом режиме, на языках ANSI-C или Visual Basic Scripts. Можно подключать динамические библиотеки DLL и работать с ActiveX объектами.

Menus & Toolbars – редактор, позволяющий создавать пользовательские меню и панели инструментов для мнемосхем и всплывающих окон.

User Administrator – удобное управление правами доступа пользователей.

Basic Process Control – набор инструментов, таких как автопостроение иерархии мнемосхем, синхронизация времени в системе, конфигурация проектов с несколькими мониторами, автопостроение экрана диагностики и др.

Коммуникация с ПЛК – в комплект поставки WinCC входит набор каналов связи для подключения к ПЛК SIMATIC S5/S7[7].

3.3.2 Дополнительный набор опций WinCC

Наличие дополнительных опций позволяет сконфигурировать систему управления под требования конкретного предприятия, при этом уменьшается стоимость базового пакета.

WinCC/Server предназначен для построения распределенных архитектур клиент/сервер. Один сервер позволяет обеспечить доступ до 32 клиентов к оперативным и архивным данным, сообщениям, мнемосхемам и отчетам через сеть Ethernet TCP/IP. В зависимости от задачи в единой системе допускается применять до 12 серверов (возможно, резервированных). В архитектуре клиент-

сервер для клиента достаточно лицензии RT128, на сервере необходима лицензия на нужное число тегов (например, RT64K) плюс опция Server. Для дублирования серверов служит опция WinCC /Redundancy (содержит 2 лицензии на каждый сервер).

WinCC/Redundancy позволяет организовать параллельную работу 2-х одноместных систем WinCC, 2-х WinCC SCADA-серверов или 2-х WinCC архивных серверов с функцией взаимного мониторинга. Каждая станция имеет свой набор связей с ПЛК и свои архивы.

При сбое в работе одной из WinCC станций управление системой передается резервной станции. В структурах клиент/сервер обеспечивается автоматическое переключение клиентов на резервный сервер, непрерывная работа оперативного управления всех WinCC станций. Механизм переключения на резервную станцию исключает возможность потери данных.

После устранения неисправностей автоматически выполняется синхронизация работы резервированной пары WinCC станций с передачей данных (данные процесса, архивы, пользовательские архивы, сообщения) из памяти работавшей станции в память включенной станции. Копирование данных в память восстановленной станции выполняется в фоновом режиме и не влияет на работу станций операторов.

WinCC/Central Archive Server (CAS) предназначен для централизованного архивирования данных всех серверов. Архивы переменных и сообщений ведутся на WinCC-серверах и по завершении сегмента передаются на CAS. С клиентов обеспечивается прозрачный доступ к данным как серверов WinCC, так и CAS. WinCC/CAS содержит все лицензии, необходимые для построения CAS, включая лицензию на 1500 архивных тегов.

WinCC/UserArchives предназначен для построения архивов пользователя с произвольной структурой и управления записью/чтением в/из них. Такие архивы могут применяться для реализации процедур рецептурного управления, хранения каких-либо связанных данных. WinCC и ПЛК (например, SIMATIC

S7) выполняют запись данных в архивы, а, при необходимости, могут обмениваться этими данными между собой. Лицензия WinCC/User Archives устанавливается только на серверы или однопользовательские станции WinCC. Пользователь получает возможность вводить параметры рецепта в WinCC, сохранять их в архиве пользователя и пересылать задания на уровень систем автоматизации. С другой стороны, ПЛК могут выполнять сбор данных в течение заданного промежутка времени (например, смены) и посылать их в пакетном виде в WinCC. Специальный ActiveX элемент позволяет производить в Runtime интерактивный просмотр содержимого архивов в табличном виде, редактирование, импорт и экспорт записей. Данные также могут быть представлены в виде свободно разрабатываемой формы.

WinCC/Web Navigator поддерживает необходимый набор функций оперативного управления и мониторинга предприятия через Internet, Intranet или локальную сеть.

Конфигурация системы на основе WinCC и WinCC/Web Navigator включает в свой состав:

- Web-сервер на основе однопользовательской станции WinCC, WinCC клиента или WinCC сервера.

- Web-клиент, выполняющий функции оперативного управления и мониторинга через Internet браузер с поддержкой элементов управления ActiveX. Базовое программное обеспечение WinCC на Web клиенте не нужно.

WinCC/DataMonitor применяется для отображения и анализа состояния техпроцесса, архивных данных и отчетов на любом офисном ПК, оснащенном стандартным набором инструментов, таких как Internet Explorer и Excel.

WinCC/DataMonitor включает в свой состав набор Internet-совместимых инструментальных средств:

- Process Screens – просмотр мнемосхем WinCC при помощи Internet Explorer (WebNavigator без управления).

– Trends&Alarms – отображение и анализ архивных данных WinCC, экспорт данных из Internet Explorer.

– Excel Workbooks – разработка отчетов и анализ данных в MS Excel. Публикация отчетов в WEB.

– Reports – запуск формирования отчетов по событию или по расписанию в форматах Excel или PDF.

– WebCenter – построение Internet портала, как центральной точки доступа ко всем данным WinCC.

IndustrialDataBridge устанавливает соединения между интерфейсами источника и приемника данных и обеспечивает передачу данных между ними.

Источники и приемники данных:

– OPC Data Access (например, WinCC, WinAC и SIMATIC NET, выступающие в роли OPC сервера);

– Базы данных SQL / OLE-DB / ODBC (MS Access, MS SQL Server, WinCC UserArchive);

– SEND/RECEIVE;

– WinCC OLE-DB Provider (только как источник);

- MS Excel, CSV файл (только как приемник).

Передача данных осуществляется циклически, по изменению или по событию.

WinCC/ODK – это набор библиотек и примеров, позволяющий использовать открытые программируемые интерфейсы для доступа к данным и функциям WinCC.

Открытые интерфейсы WinCC/ODK позволяют разрабатывать собственные приложения и дополнения для базового программного обеспечения WinCC.

SIMATIC Logon предназначен для централизованного, общепроизводственного управления пользователями, интегрированного в систему управления пользователями Windows. Администраторы могут создавать и отключать пользователей в масштабах всей компании.

WinCC/Audit реализует защищенный журнал слежения как за изменениями в конфигурации проекта, так и за действиями операторов. В систему слежения могут быть включены любые элементы интерфейса пользователя, а также таблицы UserArchive. Контроль версий проекта упрощает документирование, создание промежуточных версий проекта и предоставляет функцию отката.

WinCC/IndustrialX позволяет разрабатывать типовые ActiveX объекты, гибко конфигурируемые под требования отрасли промышленности, для которой они предназначены. Например, они могут быть адаптированы для проектов химической промышленности или машиностроительных производств.

WinCC/ProAgent позволяет создавать мощные системы диагностики и поиска неисправностей в приложениях, построенных на основе ПЛК SIMATIC S7/WinAC и устройств визуализации SIMATIC HMI. Стандартная концепция диагностики всех компонентов SIMATIC. Оптимальное взаимодействие с пакетами STEP7, S7-Pdiag, S7-Graph и системами визуализации WinCC и WinCC flexible. При возникновении неисправностей ProAgent формирует сводную и детальную информацию о месте и причинах их появления, обеспечивает поддержку ее устранения[6].

Компания Siemens является одним из лидеров по поставке оборудования в Россию, поэтому её программное обеспечение широко используется в различных отраслях производства.

3.4 SCADA-система CitectSCADA

CitectSCADA – программный продукт, представляющий собой систему мониторинга, управления и сбора данных. Интуитивные инструменты настройки и мощные функции помогают заказчикам разрабатывать и развертывать решения для приложений любого размера с надежной визуализацией и инженерными возможностями, повышающими эффективность работы, что помогает снизить риск и быстрее обеспечить эффективные действия[2].

Пример экрана SCADA системы CitectSCADA показан на рисунке 14.

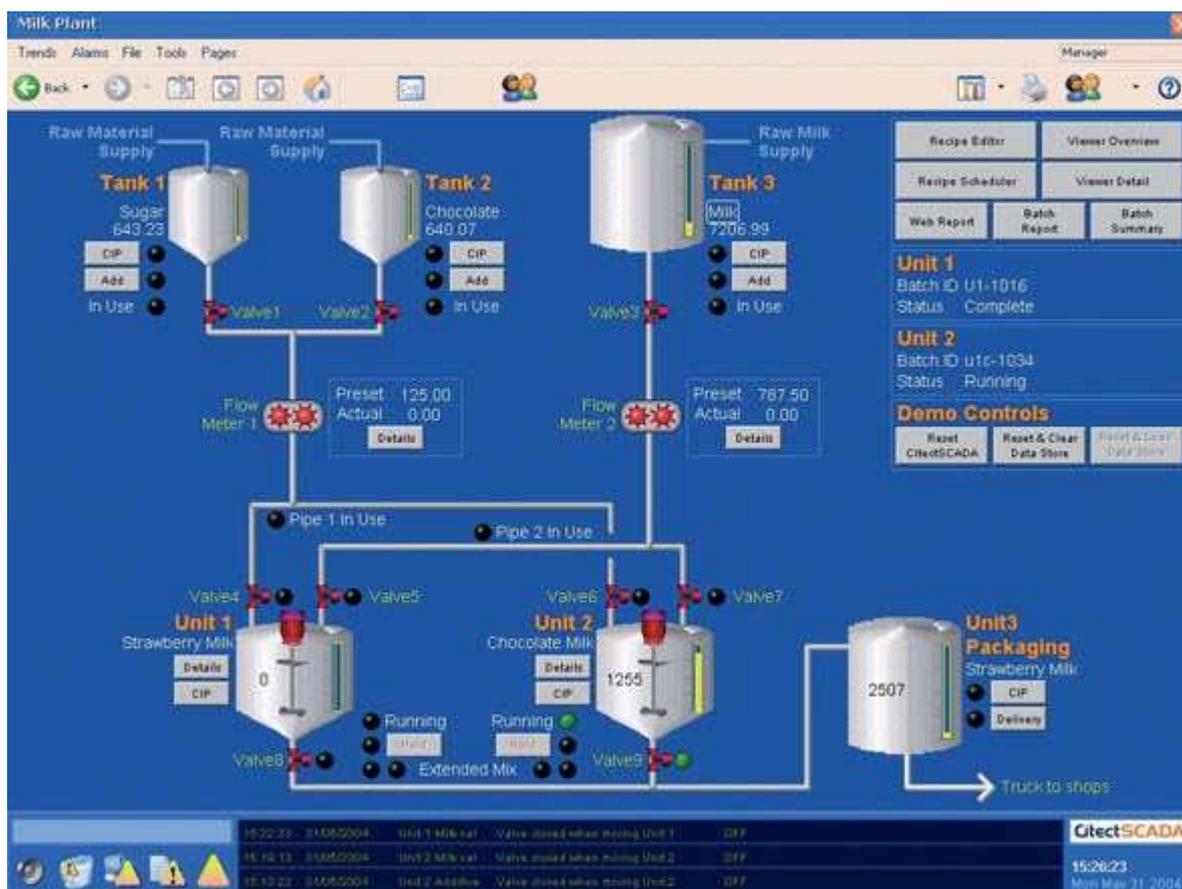


Рисунок 14 – Пример экрана SCADA системы CitectSCADA.

CitectSCADA проектировалась и разрабатывалась как средство реализации всех требований предприятия в виде единой интегрированной системы. CitectSCADA содержит все необходимые компоненты, устраняющие как необходимость использования дополнительного программного обеспечения, так и фрагментацию данных.

CitectSCADA обладает следующими характеристиками:

Высокая производительность. Высокая производительность определяется тем, что CitectSCADA построена на базе мультизадачного ядра реального времени. CitectSCADA может работать с большими объемами данных, и при увеличении количества параметров время отклика изменяется незначительно.

Модульная архитектура и масштабируемость. CitectSCADA может применяться как для небольших систем с десятками или сотнями параметров,

так и для крупных проектов с сотнями тысяч параметров. Эта масштабируемость определяется модульной клиент-серверной архитектурой, в которой каждый функциональный модуль системы может исполняться на отдельном компьютере и даже быть распределен на несколько компьютеров для увеличения общей производительности. Это означает, что если ваша система растет, то CitectSCADA может расти вместе с ней, сохраняя ваши инвестиции.

Встроенное резервирование. Многие производственные процессы не допускают остановки. В этом случае, как к аппаратуре, так и к программному обеспечению, управляющему этим процессом, предъявляются повышенные требования по надежности. Резервирование — общеизвестный способ повышения надежности компьютерных систем, но в CitectSCADA резервирование является встроенным, т.е. не требующим написания дополнительного прикладного программного обеспечения для реализации резервирования. Она допускает резервирование любого своего функционального модуля, а также каналов связи между модулями и между модулем и контроллерами ввода/вывода.

Модульная, масштабируемая архитектура клиент-сервер. CitectSCADA состоит из пяти функциональных модулей (серверов или клиентов):

– I/O — сервер ввода/вывода. Обеспечивает передачу данных между физическими устройствами ввода/вывода и остальными модулями Citect.

– Display — клиент визуализации. Обеспечивает операторский интерфейс: отображает данные, поступающие от других модулей Citect и управляет выполнением команд оператора.

– Alarms — сервер алармов (тревог). Отслеживает данные, сравнивает их с допустимыми пределами, проверяет выполнение заданных условий и отображает алармы на соответствующем узле визуализации.

– Trends — сервер трендов. Собирает и регистрирует трендовую информацию, позволяя отображать развитие процесса в реальном масштабе времени или в ретроспективе.

– Reports — сервер отчетов. Генерирует отчёты по истечению определённого времени, при возникновении определённого события или по запросу оператора.

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

Возможности человеко-машинного интерфейса. Главная задача SCADA-системы – обеспечение интерфейса оператора технологического процесса.

Для разработки операторского интерфейса CitectSCADA предоставляет библиотеки простых графических объектов (линии, фигуры, точечные изображения, текст, трубы.) и технологических символов (механизмы, резервуары, насосы).

Алармы (Alarms). Алармы – это предупреждающие сообщения. Они передаются оператору в специализированных окнах, называемых страницами или через анимируемые графические объекты, например, цвет объекта изменится с зеленого на красный при возникновении аварийной ситуации. С каждым алармом можно связать определенное действие, которое будет выполняться при появлении данного события, например, воспроизвести звуковой файл. Для работы оператора с алармами можно создать специальные справочные окна, содержащие описание действий, которые оператор должен выполнить для исправления аварийной ситуации. Информация об алармах и реакция оператора на них может выводиться как автоматически при возникновении аларма, так и по запросу оператора.

В CitectSCADA различаются четыре типа конфигурируемых разработчиком алармов:

- дискретные алармы, которые возникают при изменении состояния дискретных переменных (с 0 на 1 или наоборот);

- аналоговые алармы, которые возникают при выходе аналоговых переменных за указанные верхние и нижние пределы, при отклонении от заданного значения и при отклонении от заданного значения скорости изменения переменной;

- алармы с метками времени позволяют регистрировать сообщения с точностью до миллисекунд. Метка времени обычно используется для изучения тревожных ситуаций, когда одновременно возникает целый ряд алармов. С помощью метки времени можно выявить последовательность возникновения алармов;

- составные алармы, возникающие в результате комбинации событий.

Кроме алармов, конфигурируемых разработчиком, в CitectSCADA есть категория встроенных аппаратных/диагностических алармов. CitectSCADA регулярно запускает диагностические процедуры для проверки, как собственного состояния, так и состояния устройств ввода/вывода. Сведения об обнаруженных неисправностях сообщаются оператору автоматически. Аппаратные алармы всегда регистрируются отдельно и отображаются на отдельном дисплее алармов.

Тренды (Trends). Распределенная система построения трендов CitectSCADA может обрабатывать сотни переменных, не влияя на производительность или целостность данных. Регистрироваться и выводиться на экран может любой производственный параметр. Тренд в CitectSCADA — это изображение изменения значения какой-либо переменной (объема конечного продукта, уровня, температуры и т.д.) с течением времени, а также графическая оценка работы устройства или хода процесса. В каждый тренд может отображаться несколько переменных, которые и выводятся на экран,

давая визуальное представление о поведении процесса во времени. Необходимые выборки могут извлекаться как периодически, так и в момент возникновения в системе определенных событий. Частота выборки может изменяться от 10 миллисекунд до 24 часов.

CitectSCADA поставляется в комплекте с набором уже готовых шаблонов, обеспечивающих быстрое создание трендов, оснащенных необходимыми средствами навигации и чтения производственных параметров. Тренды могут выводиться в виде одинарных, двойных или всплывающих окон, но при необходимости вы легко сможете сконфигурировать свой собственный тренд и включить в него необходимые функции.

Отчеты (Reports). Отчет CitectSCADA – это документ, отражающий некоторые производственные показатели и выдаваемый периодически, по запросу, либо при возникновении какого-либо события (например, при изменении состояния какой-либо переменной, в момент запуска Citect или в указанное время дня). Отчеты могут генерироваться в любом удобном для пользователя формате. В него может входить форматированный текст, оперативная и накапливаемая информация, а также результаты математических вычислений. Кроме того, отчеты могут содержать и некоторые команды: замены производственных параметров, загрузки инструкций, выполнения диагностики, смены составов смесей и т.д. Отчеты могут выводиться на экран, распечатываться, а также сохраняться на диске для последующей распечатки или просмотра. Отчеты можно создавать как в текстовом формате (например *.rtf), так и формате базы данных (*.dbf). Отчет можно обрабатывать средствами любого текстового редактора и с помощью SQL-запросов. CitectSCADA допускает резервирование сервера отчетов, поэтому выдача отчетов гарантирована всегда. Если в системе работают два сервера – основной и резервный, то отчет генерируется основным. В случае его отказа отчет выдается резервным сервером. Можно сконфигурировать резервный сервер так, что он будет выдавать отчеты одновременно с основным сервером.

Развитый язык программирования Cicode. Язык Cicode, включающий более 1000 функций, позволяет создать прикладную систему любой сложности. Язык Cicode поддерживает 40 операторов для управления алармами, 19 операторов для работы с файлами, 18 SQL-функций, 50 операторов для организации работы с трендами, операторы для управления коммуникационными портами и множество других. В Cicode есть также функции для инженерной проверки производительности системы[1].

CitectSCADA получила широкое мировое распространение.

3.5 SCADA-система Trace Mode

SCADA-система Trace Mode компании AdAstra Research Group, Ltd (Россия) – это высокотехнологичная российская программная система для автоматизации технологических процессов (АСУ ТП), телемеханики, диспетчеризации, учета ресурсов (АСКУЭ, АСКУГ) и автоматизации зданий. TRACE MODE работает под Windows ® и Linux ®, используется в более, чем в 30-и странах мира, в 48-и отраслях промышленности и имеет наибольшее (49000 шт.) число реальных инсталляций в России [4].

Пример экрана SCADA системы Trace Mode показан на рисунке 15.

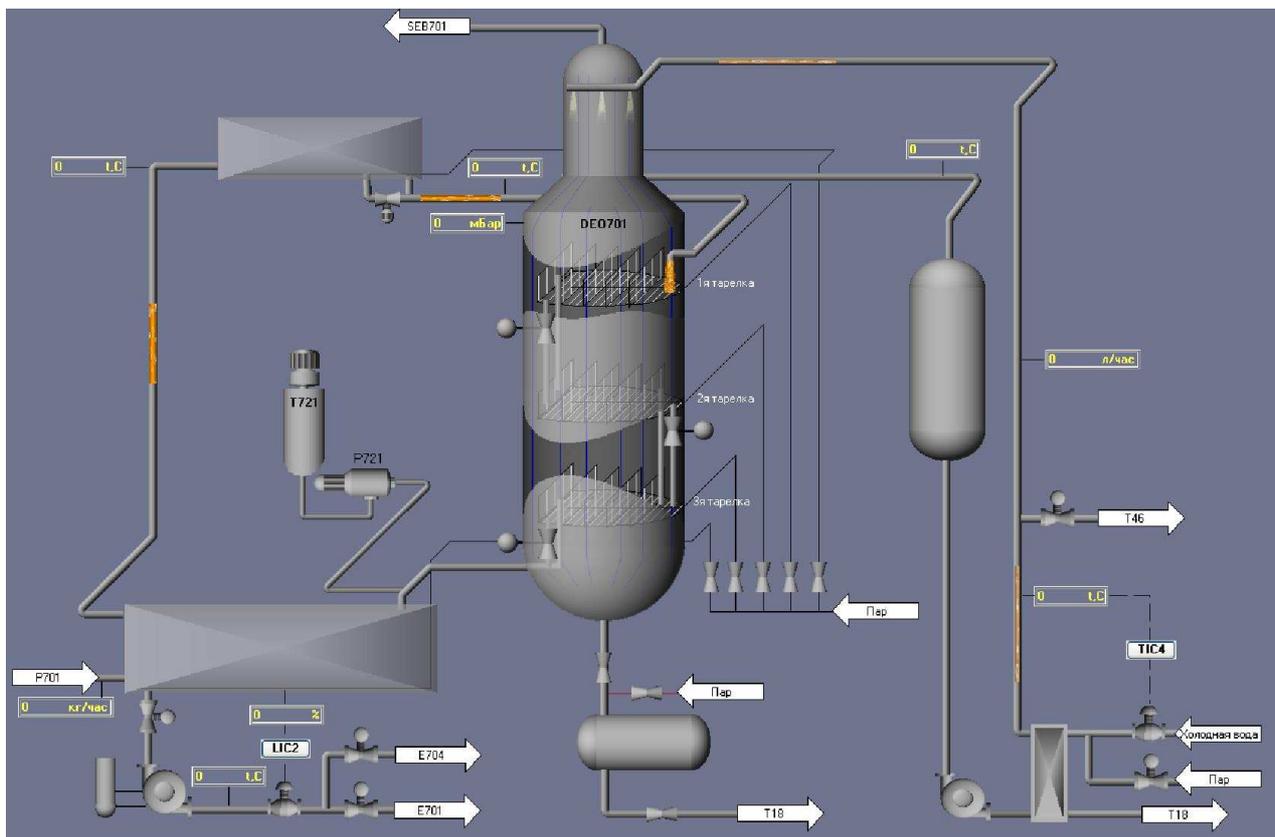


Рисунок 15 – Пример экрана SCADA системы Trace Mode.

Интегрированная платформа для управления производством Trace Mode состоит из Интегрированной среды разработки (объединяет более 10 различных редакторов проекта), в которой осуществляется создание проектов АСУ и из набора исполнительных модулей, обеспечивающих функционирование системы в реальном времени[11].

Интегрированная среда разработки включает полный набор средств построения АСУ ТП:

- операторский интерфейс (SCADA/HMI);
- промышленные контроллеры (SOFTLOGIC);
- промышленная база данных реального времени;
- система управления основными фондами и техническим обслуживанием оборудования (EAM);
- система управления производством (MES) [4].

С помощью интегрированной инструментальной системы SCADA TRACE MODE можно:

- подключиться к более чем к 2572 ПЛК, счетчикам и устройствам через бесплатные драйверы или OPC;
- разработать качественный графический операторский интерфейс (SCADA/HMI), в т.ч. и с web-доступом;
- создать базу систему записи истории процесса в собственной промышленной СУБД;
- написать программы управления на 5-и языках стандарта МЭК 6-1131/3, по расписаниям, статистическим данным или с помощью рецептов;
- настроить систему безопасности SCADA, соответствующую современным требованиям;
- запрограммировать промышленный контроллер (SOFTLOGIC) на 5-и языках стандарта МЭК 6-1131/3;
- создать систему управления тревожными и предупредительными сообщениями;
- генерировать качественные отчеты, при помощи собственного генератора;
- создать АСУ ТП в различных архитектурах: распределенных (PCU), клиент-серверных или web-ориентированных;
- разработать надежные системы в условиях плохой связи (телемеханика);
- создавать надежные резервированные системы;
- воспользоваться мощными средствами отладки и удаленной диагностики АСУ.

Достоинством программного пакета Trace Mode является большая библиотека встроенных драйверов, которая поставляется бесплатно.

Система поддерживает большое число как отечественного, так и зарубежного оборудования и позволяет разрабатывать высоконадёжные распределённые АСУ ТП.

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

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

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

Объёмные фигуры в Trace Mode обладают рядом уникальных свойств:

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

Для создания сложных мнемосхем в графическом редакторе SCADA-системы поддерживаются слои, видимостью которых можно управлять в реальном времени.

Также в системе реализована поддержка всплывающих (Pop-Up) окон. Всплывающие окна располагаются поверх окна основной мнемосхемы и служат вспомогательными элементами.

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

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

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

SCADA-система Trace Mode обладает широкими возможностями интеграции с базами данных и другими приложениями. Система поддерживает популярные программные интерфейсы: OPC, DDE, ODBC.

Взаимодействие с внешними базами осуществляется с помощью встроенного редактора SQL-запросов. Кроме того, существует возможность подключения компонентов ActiveX.

Все программы SCADA TRACE MODE делятся на 2 линии — базовую и профессиональную, несовместимые по форматам файлов проекта. Базовая и профессиональная версии TRACE MODE практически идентичны. Отличия состоят в цене – инструментальная система базовой линии бесплатна и ее можно легко скачать с официального сайта, но исполнительные модули дороже. Любой проект, разработанный в базовой версии, может быть конвертирован в профессиональную [4].

3.6 SCADA-система MasterSCADA

SCADA-система MasterSCADA – система для создания АСУТП, MES, решения задач учета и диспетчеризации объектов промышленности, ЖКХ и автоматизации зданий. MasterSCADA – самый современный, инновационный мощный и удобный инструмент для быстрой и качественной разработки систем. В нем реализованы средства и методы разработки проектов, обеспечивающие резкое сокращение трудозатрат и повышение надежности создаваемой системы. Разрабатывать проекты в Master SCADA легко и приятно, в ней реализован объектный подход к разработке систем управления, учета или диспетчеризации [5].

Пример экрана SCADA системы MasterSCADA показан на рисунке 16.



Рисунок 16 – Пример экрана SCADA системы MasterSCADA.

Общее число внедрений MasterSCADA системы составляет десятки тысяч практически во всех отраслях промышленности в России, ближнем и дальнем зарубежье.

Основные преимущества MasterSCADA.

Единая среда разработки SCADA-системы. Разработка всех элементов проекта MasterSCADA ведется в единой инструментальной среде. Это и разработка аппаратной архитектуры системы, и проектирование алгоритмов обработки, описание логической структуры и базы параметров, настройка первичной обработки, разработка системы визуализации, отчетов и журналов и прочих компонентов системы.

Такой подход, когда все инструменты находятся под рукой, а проект представляет единое информационное пространство позволяет легко решать следующие задачи:

- решить проблемы программной стыковки различных устройств системы управления;
- перераспределять сигналы или алгоритмы их обработки по отдельным устройствам;
- создавать распределенные по устройствам алгоритмы контроля и управления;
- иметь доступ с любого рабочего места к любой информации, имеющейся в системе.

Двухслойная структура проекта. В MasterSCADA проект разделен на два слоя:

- слой описания аппаратной структуры проекта. Здесь описываются рабочие станции, контроллеры и модули ввода-вывода, входящие в проект. Задаются их свойства и настраиваются связи между ними;
- слой описания логики проекта. Здесь описывается логическая структура проекта: какие технологические объекты автоматизируются в данном проекте, их свойства, параметры, алгоритмы управления и мнемосхемы. Кроме того,

здесь разрабатываются отчеты, настраиваются журналы и параметры информационной безопасности системы.

Такой подход дает возможность:

- разрабатывать эти структуры параллельно;
- независимо работать специалистам различных профилей;
- решить проблему перехода от одной технической структуры системы к другой (например, реализовать тот же проект на базе других контроллеров).

Открытость и следование стандартам. MasterSCADA – это максимально открытый продукт. Это достигается максимальным использованием стандартных интерфейсов и спецификаций. Под открытостью понимается:

- взаимодействие с другими программами с помощью современных технологий (OPC, OLE, DCOM, ActiveX, OLE DB, ODBC и др.);
- использование в операторском интерфейсе системы документов любого типа (например, WORD или EXCEL) и обмен данными с ними;
- неограниченное расширение функциональности MasterSCADA продуктами сторонних разработчиков;
- открытая объектная модель и возможность добавления пользовательских алгоритмов или функций на языке C++ или C#;
- связь с АСУ производством;
- открытые интерфейсы для создания пользователем любых базовых элементов.

Удобство инструментария. MasterSCADA обладает следующими особенностями:

- простой и понятный русскоязычный интерфейс;
- реализация большинства действий пользователя методом «перетащи и брось»;
- подробный справочный материал;
- интерактивный мультимедийный обучающий курс;

- контекстная справка;
- всплывающие подсказки;
- контроль допустимости вводимой информации;
- соответствие проекта логике восприятия системы и объекта разработчиком;
- возможность полной отладки проекта без связи с объектом;
- возможность полной отладки распределенной системы на одном компьютере;
- отсутствие необходимости настройки сети или выделения отдельного сервера для запуска распределенной системы;
- возможность многократного использования любой ранее созданной части проекта.

Обширные библиотеки. MasterSCADA содержит следующие библиотеки:

- библиотека объемных элементов со встроенным индикатором уровня заполнения;
- библиотеки технологических объектов из различных областей;
- библиотека алгоритмов стандарта IEC 61131-3;
- библиотека алгоритмов OSCAT (более 300 алгоритмов).

Неограниченная гибкость вычислительных возможностей. В MasterSCADA реализованы следующие вычислительные возможности:

- в MasterSCADA поддерживаются языки стандарта IEC 61131-3;
- первичная обработка каждого сигнала с автоматическим контролем до 6 границ;
- формульные вычисления значений и событий с обширной библиотекой функций;
- автоматическая и пользовательская обработка признаков качества значений;
- автоматическая индикация значений всех вычисленных сигналов;

- имитационный режим с индивидуальным выбором функций имитации сигналов;
- возможность создания пользовательских функциональных блоков и макроблоков;
- интеграция вычислительных, событийных и визуальных функций объектов;
- открытая объектная модель и возможность подключения скриптов, написанных на "С#";
- открытые интерфейсы для подключения собственных алгоритмов, написанных на языке "С";

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

Демо версии. Необходимым условием выбора SCADA-системы для разработки проекта является возможность познакомиться с ней в деле. Без этого разработчик не сможет определить устраивает его данный инструмент или нет. Поэтому для всех модулей MasterSCADA имеются демонстрационные версии.

В настоящий момент поддерживается две версии MasterSCADA.

MasterSCADA 3.X – самая популярная отечественная SCADA-система. Популярность MasterSCADA подтверждена оценками многих экспертов и опросами на профильных порталах в Интернет. Так, например, MasterSCADA признана Продуктом Года по выбору русской редакции авторитетного международного журнала Control Engineering. На базе MasterSCADA 3.x реализовано более 10000 внедрений. Среди реализованных проектов – глобальные системы с более чем 100000 параметров, приходящих на один сервер опроса, и с более чем 300 местами операторов.

MasterSCADA 4D – это продукт нового поколения SCADA-систем. В нем, по сравнению с предыдущей версией, существенно расширены инструменты по созданию крупных распределенных систем с возможностью использования технологий Интернета вещей, повышено удобство и гибкость, расширены возможности использования различных аппаратных платформ и операционных систем, увеличено число поддерживаемых уровней систем управления и реализована миграция функционала между уровнями.

В *MasterSCADA* легко разрабатывать проекты любого масштаба и сложности. Для этого предлагаются различные подходы, обеспечивающие наиболее комфортные условия разработки под каждый тип проекта [5].

Заключение

Применение интегрированной информационной системы управления предприятием позволяет существенно повысить эффективность производства. Внедрение подобной системы сложный трудоёмкий процесс, требующий волевых решений и полного понимания всех стадий внедрения со стороны руководства предприятия.

В учебном пособии особое внимание уделено уровню автоматизации технологических процессов. На этом уровне применяются SCADA-системы.

Приведены основные характеристики SCADA-систем.

Рассмотрены наиболее известные в России системы от ведущих производителей программного обеспечения систем автоматизации.

Большинство SCADA-систем в той или иной мере поддерживают языки международного стандарта МЭК 6-1131/3:

FBD (Function Block Diagram) – язык функциональных блоковых диаграмм;

LD (Ladder Diagram) – язык релейных или релейно-контактных схем;

SFC (Sequential Function Chart) – язык последовательных функциональных схем;

ST (Structured Text) – текстовый язык программирования высокого уровня;

IL (Instruction List) – машиноориентированный текстовый язык программирования.

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

В пособии приведён обзор популярных в России SCADA-систем:

SCADA-система InTouch ;

SCADA-система Genesis32;

SCADA-система WinCC;

SCADA-система CitectSCADA;

SCADA-система Trace Mode;

SCADA-система MasterSCADA.

Показаны характеристики и особенности данных SCADA-систем.

Список литературы

1. CitectScada – программное обеспечение автоматизации технологических процессов [электронный ресурс]. – Режим доступа: <http://powergroup.com.ua/citectscada>. – (Дата обращения: 04.08.2017).
2. CitectSCADA Supervisory Control and Data Acquisition (SCADA) software solution [электронный ресурс]. – Режим доступа: <http://software.schneider-electric.com/products/citect-scada/>. – (Дата обращения: 04.08.2017).
3. IL (Instruction List) — Национальная библиотека им. Н. Э. Баумана [электронный ресурс]. – Режим доступа: [http://ru.bmstu.wiki/IL_\(Instruction_List\)](http://ru.bmstu.wiki/IL_(Instruction_List)). – (Дата обращения: 10.08.2017).
4. SCADA TRACE MODE [электронный ресурс]. – Режим доступа: <http://www.adastra.ru/products/dev/scada/>. – (Дата обращения: 04.08.2017).
5. SCADA СИСТЕМА MASTERSCADA [электронный ресурс]. – Режим доступа: <http://insat.ru/products/?category=9>. – (Дата обращения: 05.08.2017).
6. SCADA система SIMATIC WinCC — основа для построения «Интеллектуального производства» [электронный ресурс]. – Режим доступа: http://www.ste.ru/siemens/pdf/rus/08_WinCC_62_r.pdf. – (Дата обращения: 04.08.2017).
7. SCADA-система SIMATIC WinCC V7.3 [электронный ресурс]. – Режим доступа: http://www.ste.ru/siemens/pdf/rus/WinCC_V73.pdf. – (Дата обращения: 04.08.2017).
8. Wonderware InTouch [электронный ресурс]. – Режим доступа: <http://www.wonderware.ru/hmi-scada/intouch/>. – Wonderware InTouch – (Дата обращения: 03.08.2017).
9. Дедюхина М. А. Информационная поддержка контроллинга на основе ERP-систем // Вестник Удмуртского университета. Серия «Экономика и право». 2010. №4. С.22-27

10. Загидуллин Р. Р., Управление машиностроительным производством с помощью систем MES, APS, ERP. — Старый Оскол: ТНТ, 2011. — 372 с.
11. Интегрированные системы проектирования и управления: SCADA-системы : учебное пособие / И. А. Елизаров, А. А. Третьяков, А. Н. Пчелинцев и др. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 160 с.
12. Минаев И. Г. Программируемые логические контроллеры в автоматизированных системах управления / И. Г. Минаев, В. М. Шарапов, В. В. Самойленко, Д. Г. Ушкур. 2-е изд., перераб. и доп. - Ставрополь: АГРУС, 2010. - 128 с.
13. Миронов А.А., Мордвинов В.А., Скуратов А.К. Семантико-энтропийное управление OLAP и модели интеграции xOLAP в SemanticNET (ONTONET). Информатизация образования и науки №2, 2009. С. 21-30.
14. Мишунин, В.В., Информационно-измерительные и управляющие систем: Учебно- методическое пособие / В.В. Мишунин, Е.В. Корсунова, В.И. Ищенко, А.В. Курлов. – Белгород: Изд-во БелГУ, 2010. – 129 с.
15. Петров И.В. Отладка прикладных ПЛК программ в CoDeSys // Промышленные АСУ и контроллеры. 2006. № 2. С.35-39.
16. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования/Под ред. проф. В.П. Дьяконова. – М.: СОЛОН+Пресс, 2004.
17. Полянсков Ю. В., Интеграция САPP-, PDM-, ERP-систем в единое информационное пространство производственного предприятия / Полянсков Ю. В., Кондратьева А. С., Черников М. С., Блюменштейн А. А. // Известия Самарского научного центра РАН. 2013. №4-3. С.628-633
18. Солдатов С., Интеграция SCADA-систем и систем управления предприятием// СТА. 2016. №1. С. 90-94
19. Швецов Д., SCADA-система GENESIS32 в сквозной автоматизации производства // ИСУП. 2007. № 4. С.7-12.

20. Язык программирования FBD [электронный ресурс]. – Режим доступа:
<http://autoworks.com.ua/programmirovanie-kontrollerov/yazyk-programmirovaniya-fbd/>. – (Дата обращения: 06.08.2017).

Электронное учебное издание

Алексей Александрович **Силаев**
Елена Юрьевна **Силаева**
Максим Алексеевич **Трушников**

**Интегрированные системы проектирования и управления
автоматизированных и автоматических производств**

Учебное пособие

Электронное издание сетевого распространения

Редактор Н.М. Матвеева

Темплан 2018 г. Поз. № 30.

Подписано к использованию 19.02.2018 г. Формат 60×84 1/16.

Гарнитура Times. Усл. печ. л. 4,63.

Волгоградский государственный технический университет.
400005, г. Волгоград, пр. Ленина, 28, корп. 1.

ВПИ (филиал) ВолгГТУ.
404121, г. Волжский, ул. Энгельса, 42а.