



Применение UltraLogic в проектировании систем управления инженерным оборудованием

Борис Шпиз, Борис Якубович, Валерий Журавлев, Ренат Биусов,
Сергей Шакиров

На примере автоматизации крупного теплового пункта рассматривается практическое применение инструментального пакета UltraLogic.

Современные большие здания представляют собой достаточно сложные инженерно-технические сооружения. Системы вентиляции, отопления, водоснабжения, канализации, лифтовое хозяйство и освещение являются объектами, вызывающими непреходящую головную боль служб эксплуатации. Управляющие контроллеры в данной ситуации — неутомимые труженики, дено и ночью следящие за порядком в доме. В этой статье показаны некоторые возможности инструментального пакета UltraLogic в решении вопросов автоматизации систем жизнеобеспечения. Материал иллюстрирован фрагментами реальных проектов, внедренных в ряде административных зданий Москвы.

UltraLogic реализован в соответствии с требованиями стандарта IEC 1131-3 и предназначен для разработки прикладного программного обеспечения сбора данных и управления технологическими процессами, выполняемыми на программируемых контроллерах с открытой архитектурой. В качестве языка программирования в UltraLogic реализован язык функцио-

нальных блоковых диаграмм (Function Block Diagram, FBD), предоставляющий пользователю механизм объектного визуального программирования (см. «СТА» № 3/97).

Введение

Итак, основу микроклимата в зданиях определяют системы теплоснабжения и вентиляции. Эти системы имеют большое количество территориально-распределенных датчиков температуры и давления, управляемых заслонок, электродвигателей и насосов. В задачи теплоснабжения входит поддержание постоянной температуры и давления в системах горячего водоснабжения при непрерывно меняющемся расходе воды. В

холодное время года температура воды в отопительных системах является функцией от температуры наружного воздуха. На тепловых пунктах имеется резервное оборудование, которое немедленно включается в работу при аварии на работающем оборудовании. Системы вентиляции зданий должны обеспечивать приток и вытяжку воздуха, его подогрев или охлаждение в зависимости от температуры наружного воздуха.

Рассмотрим типовую архитектуру системы управления жизнеобеспечением здания.

Архитектура

Как правило (рис. 1), система содержит два уровня управления: диспетчерский и технологический, которые состоят, соответственно, из центрального диспетчерского пульта (ЦДП) и группы программируемых логических контроллеров (ПЛК), объединенных локальной вычислительной сетью. ПЛК непосредственно управляют оборудованием, осуществляют съем параметров датчиков, принимают с ЦДП команды и отправляют на него значения контролируемых параметров.

В качестве ЦДП используется один или несколько персональных компьютеров IBM PC, на которых установлено программное обеспечение операторских станций АСУ ТП.



Здание Российской Академии госслужбы, где внедрена система управления инженерным оборудованием на базе UltraLogic

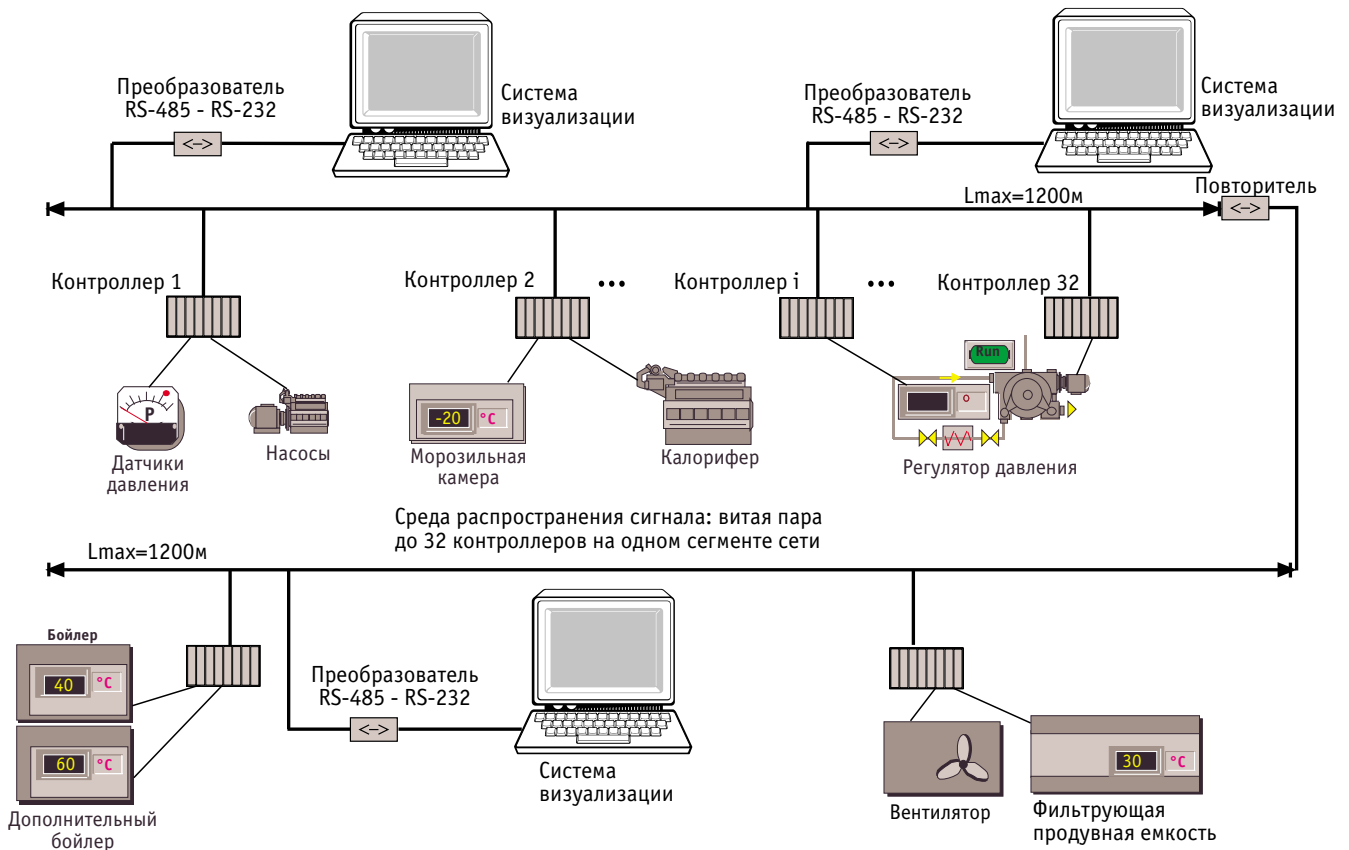


Рис. 1. Типичная двухуровневая система управления оборудованием жизнеобеспечения зданий

Программное обеспечение верхнего уровня выполняет следующие функции:

- визуализация мнемосхемы и значений параметров объекта;
- анимация состояний агрегатов и механизмов;
- прием команд диспетчера об уставках процесса и режимах работы, пересылка их на ПЛК;
- предупреждение диспетчера о предаварийной и аварийной ситуации на объекте;
- ведение осциллограмм процессов;
- ведение отчетов, архивов событий и аварийных сообщений.

В качестве системы визуализации может быть использован любой SCADA-пакет, например Genesis. UltraLogic легко подключается к SCADA посредством DDE-протокола или с помощью специального поставляемого драйвера.

UltraLogic использует метод сетевого взаимодействия между контроллерами и системой визуализации данных. В сети могут быть активные участники — Master и пассивные — Slave. Общее количество узлов сети может быть 255, каждый из которых имеет свой уникальный сетевой адрес (имя). Переменные, участвующие в сетевом обмене, имеют двойное имя, состоящее из имени переменной и префикса, являющегося сетевым адресом (или именем) уз-

ла. Для SCADA они представляются как переменные единого технологического процесса безотносительно территориального расположения контроллеров.

Контроллеры могут работать как в сетях, использующих коммуникационные порты процессора (предпочтительно использовать интерфейс RS-485), так и в сетях Arcnet, Ethernet.

Построение многоуровневых систем является следствием разнородности задач, выполняемых на каждом уровне управления. Нижний уровень (контроллеры) работает в непрерывном круглосуточном режиме, скрыт от людских глаз толщей подвалов и чердаков, реализует функции управления в «жестком» реальном времени.

Верхний уровень управления (ЦДП) работает под управлением ОС Windows 95. Это нерасторопное детище Microsoft ставит свои собственные интересы выше каких-то там внешних событий, может «уйти в себя» или радостно сообщить, что ваше приложение «зависло». Тем не менее большинство SCADA-систем работает под Windows, обеспечивая интерфейс с оператором в виде красивых мультимедийных картинок, кнопочек, графиков и таблиц. По изложенным причинам на компьютер ЦДП не возлагается никаких задач алгорит-

мического управления. Он может быть выключен и включен в любой момент времени, не прерывая и не нарушая процесса управления объектом. На нем можно запускать другие задачи пользователя (например, потихоньку сыграть в Tetris, пока начальства рядом нет), при этом контроллеры будут работать, как ни в чем не бывало.

Программирование нижнего уровня управления

UltraLogic состоит из двух компонентов: системы программирования, работающей на IBM PC совместимом компьютере, и системы исполнения, работающей на целевом контроллере. Система программирования UltraLogic функционирует в операционной среде Windows. Целевая программа загружается под управлением штатного загрузчика вычислительного модуля контроллера и для своей работы не требует ресурсов его операционной системы. Таким образом, она несет полную ответственность за свои действия, решительно отвергая услуги BIOS, исключая этим неоднозначность своего поведения на оборудовании разных производителей.

Итак, рассмотрим основные этапы создания программы, которые состоят в следующем.

1. Заполнение таблиц глобальных переменных.

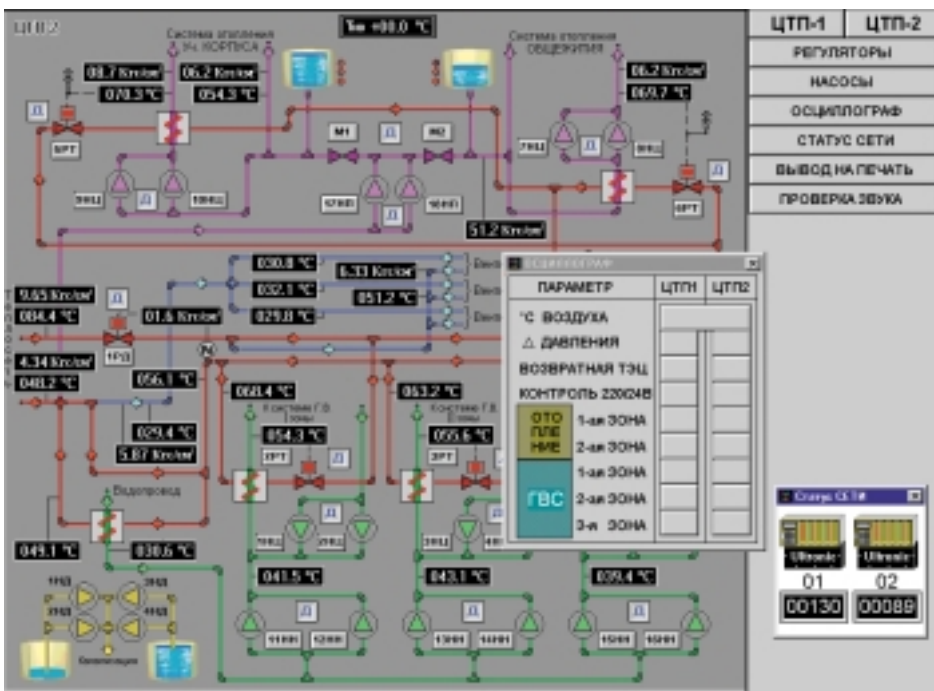


Рис. 2. Фрагмент мнемосхемы системы управления теплоснабжением

2. Конфигурирование контроллера.
3. Привязка переменных ко входам и выходам контроллера.
4. Разработка алгоритмов программ.
5. Компиляция.
6. Загрузка исполняемого кода в контроллер.
7. Отладка программы.

Заполнение таблиц глобальных переменных

Переменные в проекте являются главными действующими лицами в работе всей системы, они как живая сила на поле боя. Через переменные выполняются команды включить-выключить, принять значение с датчика, переслать информацию на верхний уровень и т. п.

Переменные хранятся в специальной базе данных и могут иметь следующие атрибуты:

- константа (Constant) — переменная, значение которой устанавливается один раз на этапе разработки программы;
- входная (Input) — переменная, которая может быть привязана к одному из входов контроллера;
- выходная (Output) — переменная, которая может быть привязана к одному из выходов контроллера;
- сетевая (Network) — переменная, значение которой передается сетевому драйверу, входящему в состав системы исполнения, для последующей передачи по сети на базе RS-485 или Ethernet, объединяющей



Тепловой пункт

контроллеры и верхний уровень системы.

Поместив переменную в список Network, мы делаем ее доступной сетевому драйверу. Поместив переменную в список Input, мы заставим программу в каждом цикле считывать значение переменной с одного из входов контроллера, привязанного к данной переменной. Поместив переменную в список Output, мы заставим программу в каждом цикле помещать значение переменной на один из выходов контроллера, привязанного к данной переменной. Разбиение переменных на списки носит условный характер и служит для удобства и структуризации данных. Одна и та же переменная может находиться в разных списках. Переменные сортируются по любым возможным признакам, легко организован поиск и редактирование. Имена программам и переменным присваивают по следующим правилам:



Два контроллера заменили старую систему управления тепловым пунктом



Так выглядела старая система управления тепловым пунктом

- имя не может содержать более 255 символов;
- первый символ должен быть буквой или символом подчеркивания, имя переменной может содержать буквы латинского и русского алфавита, а также символы подчеркивания.

Возможность вводить длинные имена на русском языке делает проект самодokumentируемым, исключает необходимость ввода условных обозначений и неоднозначного толкования имен.

Проект теплового пункта, мнемосхема которого приведена на рис. 2, содержит 537 глобальных переменных, из которых примерно половина — сетевые. Фрагмент мнемосхемы управления вентиляцией представлен на рис. 3. Типовая схема управления регулятором температуры дана на рис. 4.

Конфигурирование контроллера

Конфигурирование контроллера включает в себя операции по выбору аппаратной платформы, типа вычислительного модуля, модулей ввода/вывода. В качестве аппаратной платформы UltraLogic предлагает платформу MicroPC (Octagon Systems) или другие, например платформу ADAM 5000 (Advan-

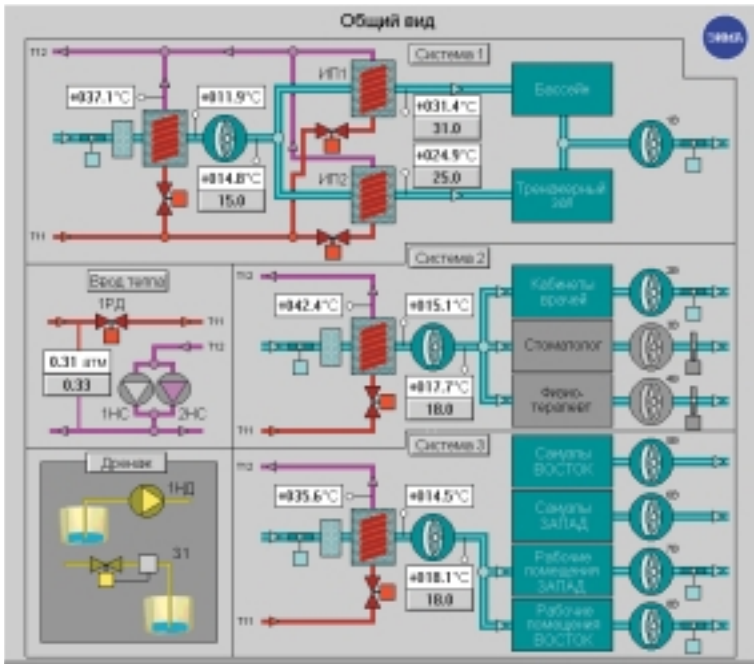


Рис. 3. Фрагмент мнемосхемы управления вентиляцией

tech). Конфигурирование делается так, как будто вы заполняете анкету. В специальных последовательно возникающих окнах галочками отмечаете то, что вам требуется в контроллере: тип вычислительного модуля, наличие локальной сети и сторожевого таймера, номенклатуру модулей ввода/вывода и их количество. Эта информация будет использована компилятором проекта. На этом конфигурирование заканчивается.

Следует заметить, что программы, собранные для платформы MicroPC, могут исполняться на других персональных IBM PC совместимых компьютерах, оборудованных периферией, описанной в конфигурации.

На рис. 5 приведен общий вид окна UltraLogic с загруженным проектом.

Привязка переменных проекта

Входы и выходы контроллера являются входами и выходами тех модулей, из которых сконфигурирован контроллер. Данный этап предполагает, что перед вами лежит электрическая схема подключения контроллера к объекту и необходимо назначить каждому входу/выходу имя переменной, с которым будет опе-

ризовать программа. По области видимости в рамках программы, созданной на UltraLogic, переменные делятся на глобальные (public) и локальные. Глобальные переменные доступны во всех FDB-программах и внешних процедурах, входящих в программный модуль. Локальные переменные доступны только в пределах одной FDB-программы. В специальных окнах-трафаретах, соответствующих выбранным вами модулям,

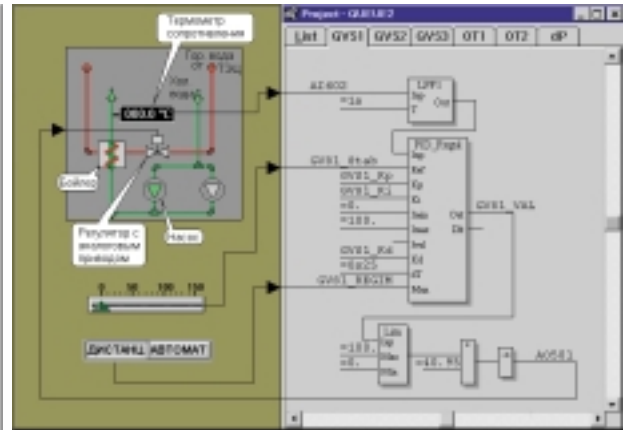


Рис. 4. Типовая схема управления регулятором температуры

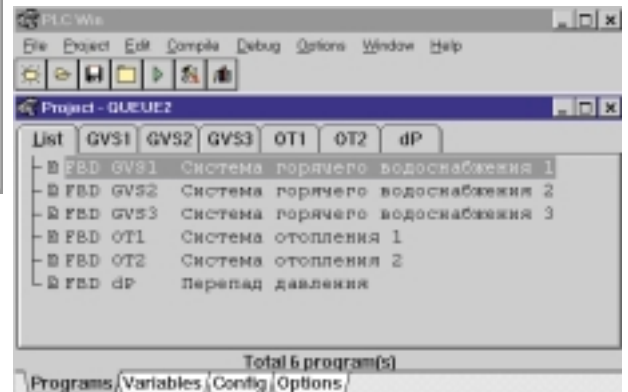


Рис. 5. Общий вид окна UltraLogic с загруженным проектом

UltraLogic предлагает для привязки имени по умолчанию, ассоциированные с географическим расположением входа/выхода в контроллере (рис. 6). Так, например, для первого входа модуля дискретного ввода, установленного на первом месте, будет предложено имя DI101, а для пятого выхода модуля

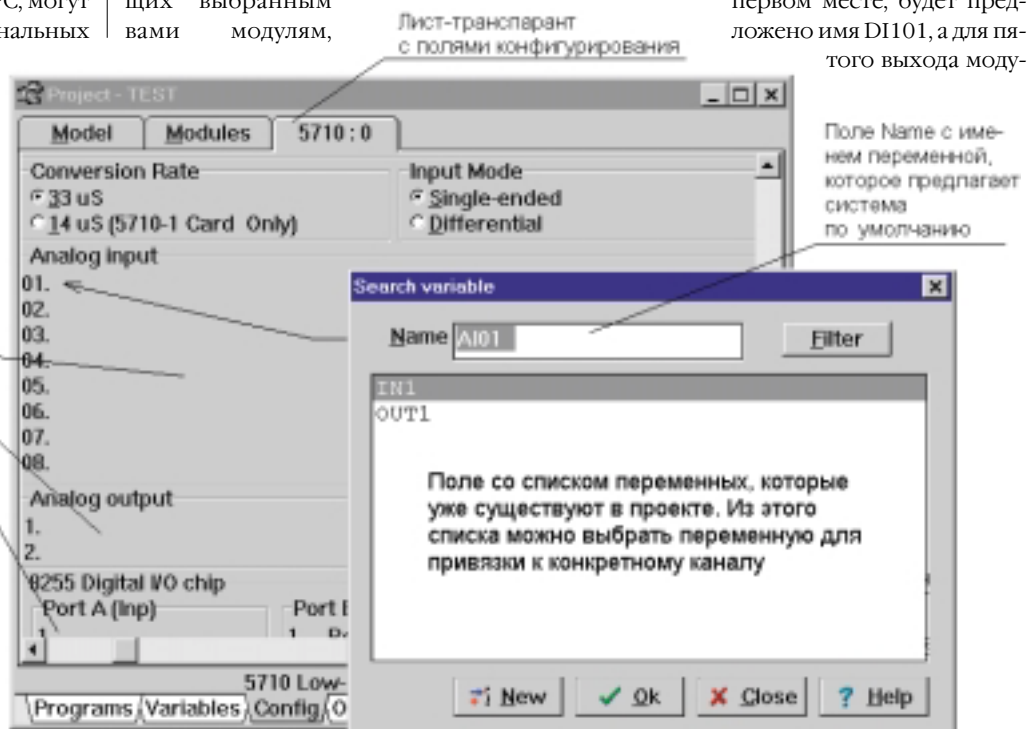


Рис. 6. Типовое окно привязки переменных

ля аналогового вывода, установленного на втором месте, будет предложено имя АО205. Если вас не устраивает это имя, UltraLogic предложит выбрать имя из таблиц глобальных переменных, при этом его не надо переписывать вручную, а просто достаточно указать мышью в таблице. Наконец, вы можете ввести новую переменную для любого входа/выхода, при этом конфигурактор немедленно поместит эту переменную в таблицы глобальных переменных.

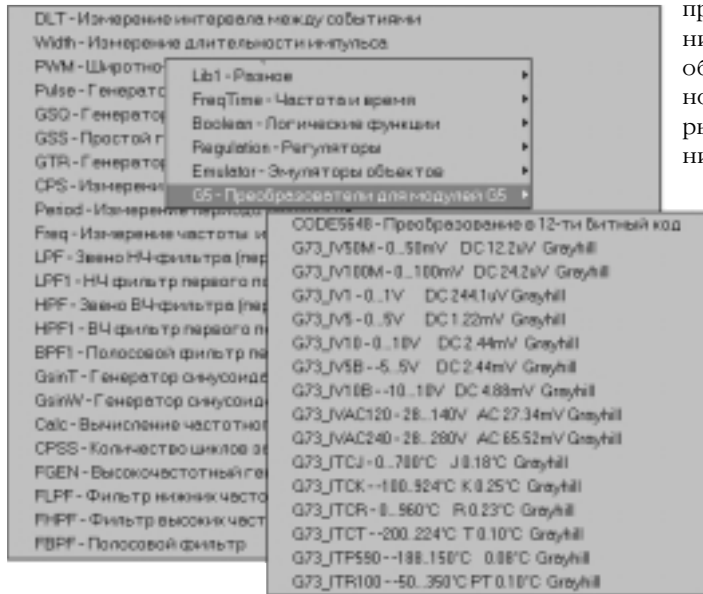


Рис. 7. Окно выбора библиотечных блоков

Разработка программ

Непосредственный процесс разработки программ в UltraLogic заключается в «сборке» с помощью мыши программы из готовых «кубиков» — функциональных блоков (рис. 7). «Сборка» осуществляется в среде специального графического редактора. Редактор содержит такие операции, как вызов из библиотек функциональных блоков, копирование, перемещение и удаление объектов, ввод связей между объектами, ввод комментариев. Редактор позволяет производить групповые операции, масштабировать изображение. Наличие многооконного интерфейса позволяет организовать работу таким образом, что перед глазами программиста находится вся необходимая информация. Возможность переносить информацию между окнами с помощью мыши по существу исключает ошибки, связанные с повторным вводом имен, типов переменных и т. п. Редактор имеет весьма интересную функцию — построение иерархического дерева программы, показывающего последовательность выполнения функциональных блоков программы. Иерархическое дерево (рис. 8) позволяет судить о том, насколько ваше собственное понимание порядка выполнения элементов программы совпадает с мнением компилятора по данному вопросу. Кроме этого, редактор позволяет исправлять существующие библиотечные блоки, добавлять собственные.

Библиотеки, поставляемые в составе пакета UltraLogic, охватывают весьма широкий спектр алгоритмов автоматического управления, обработки сигналов, приложений для работы с первичными преобразователями различных

фирм. Все библиотеки являются открытыми, с подробным описанием работы алгоритма и временных характеристик. Пример библиотечного блока фильтра нижних частот дан на рис. 9.

Работа системы исполнения имеет циклический характер. Циклы контроллера UltraLogic организует следующим образом: сначала считываются все входы контроллера в глобальные переменные, имеющие атрибут Input, затем осуществляется один проход всех программ, после чего происходит вывод всех глобальных переменных, имеющих атрибут Output, на соответствующие (привязанные к ним) выходы модулей контроллера. Во время прохода программ значения входных и выходных переменных остаются неизменными. Весь контролируемый технологический процесс разбивается на ряд формальных независимых задач, каждая из которых обслуживается отдельной программой. Программы выполняются поочередно с одинаковым приоритетом. Если время одного цикла системы исполнения существенно меньше скорости изменения параметров управляемого процесса, можно считать, что все программы выполняются параллельно, т. е. одновременно.

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 заиклиться на неопределенное время, например на ожидании какого-либо события. Данное правило является «хорошим тоном» в программировании на языке FBD и гарантирует, что никакая

программа не приостановит исполнение других программ. Такой подход обеспечивает простоту и прозрачность для понимания. Так как нет прерываний, система полностью детерминирована. Задачи вызываются в одной

и той же последовательности, что позволяет достаточно просто произвести анализ «наихудшего случая» и вычислить максимальную задержку. Так, контроллер с вычислительным модулем 5012A (IBM XT) на тепловом пункте ЦТП-2 (рис. 2) одновременно обслуживает 6 независимых технологических процессов с быстродействием 130-150 циклов в секунду, при этом успевая обмениваться данными с системой визуализации со скоростью 57 кбод. Такого быстродействия более чем достаточно, чтобы считать, что

все процессы на тепловом пункте обслуживаются одновременно. (Заметим в скобках, что производительность вычислительного модуля 5066 в несколько тысяч раз выше, чем модуля 5012A.)

На рис. 4 приведен фрагмент программы, реализующей алгоритм управления регулятором температуры. Сигнал AI602 с датчика температуры поступает на фильтр нижних частот LPF1 с постоянной времени фильтра $T = 1$ с. Отфильтрованное замеренное значение температуры воды в зоне отопления сравнивается с заданным значением GVS1_Stab. При этом выставляется

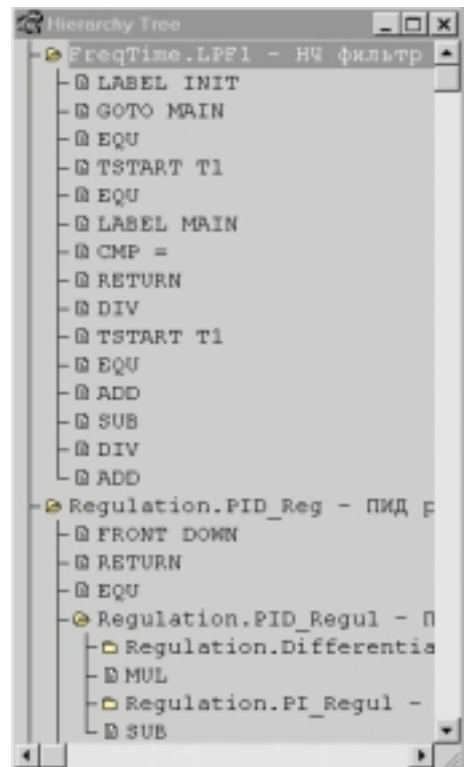


Рис. 8. Иерархическое дерево программы

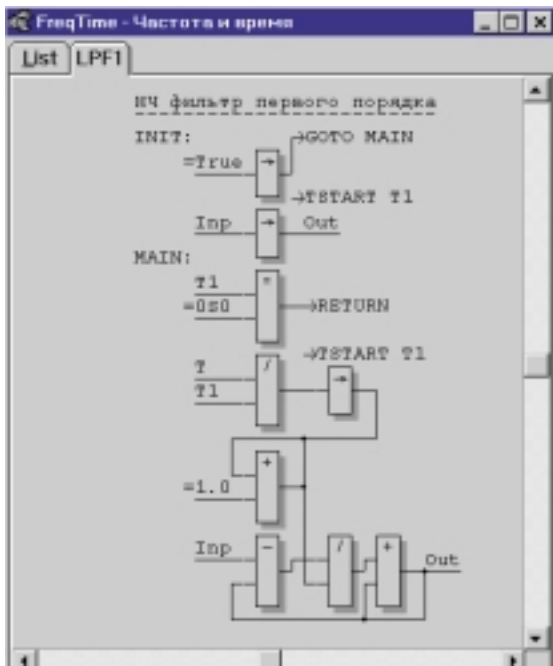


Рис. 9. Библиотечный блок фильтра нижних частот (LPF).
Постоянная времени задается переменной T

такое значение регулирующего воздействия GVS1_Val, чтобы эти температуры были равны. Для избежания автоколебаний в системе управление осуществляется в соответствии с законом ПИД-регулирования. Коэффициенты пропорционального GVS1_Kp, интегрального GVS1_Ki и дифференциального регулирования GVS1_Kd формируют переходную характеристику контура регулирования и подбираются опытным путем на этапе отладки. Регулирующее воздействие GVS1_Val ограничивается пределами 0-100% и переводится непосредственно в код 12-разрядного цифро-аналогового преобразователя. Последний функциональный блок программы, который формирует выходной сигнал регулирования АО501, предназначен для преобразования типа переменной из формата Float в формат Integer. Температура в зоне отопления задается вручную при дистанционном управлении либо как функция температуры наружного воздуха при автоматическом управлении. В приведенной программе функциональные блоки LPF1 и Pid_Regul являются библиотечными элементами, а все остальные — базовыми функциональными блоками.

Алгоритмы управления всеми видами технологического оборудования могут инкапсулироваться в библиотечные блоки и комбинироваться в дальнейшем произвольным образом. Хорошо продуманные алгоритмы включают в себя перечень операций, которые необходимо выполнить для аварийной остановки неисправного оборудования и

локализации последствий аварии, после чего управление группой оборудования, где произошла авария, передается оператору. Предусматривается также возможность проведения регламентных работ, технического обслуживания и ремонта оборудования.

Компиляция проекта

Программа из «кубиков» собрана, необходимо ее превратить в команды конкретного процессора. Для этого специальной кнопкой на панели инструментов запускается компилятор, который из графического образа программы создает объектный файл. Настройки компилятора осуществляются автоматически на этапе конфигурирования вычислительного модуля контроллера. Компилятор проверяет программу и в случае обнаружения ошибок выдает в специальное окно соответствующие сообщения. Для локализации ошибки в программе достаточно щелкнуть кнопкой мыши на строке сообщения об ошибке. Редактор автоматически откроет программу с данной ошибкой и выделит красным цветом объекты, где она обнаружена.

Отладка программ

Программа готова. Ее можно опробовать, не загружая в контроллер. UltraLogic может эмулировать работу контроллера непосредственно на компьютере системы программирования. Этот режим обычно используется для первоначальной отладки алгоритмов, а также весьма полезен в учебном процессе. Эмулятор контроллера позволяет отлаживать программу в непрерывном и пошаговом режиме, задавать точки останова, просматривать списки всех переменных, в том числе и локальных.

Практическая отладка программы управления сложным объектом является достаточно трудоемким процессом и сопряжена с опасностью непредвиденной реакции реального объекта на управляющие воздействия. Избежать подобной ситуации, а также произвести предварительный подбор управляющих воздействий позволяет моделирование объекта управления. Модель объекта управления представляет из себя отдельную программу или набор программ. Эти программы принимают от управляющей программы сигналы управления в виде переменных и моделируют поведение объекта, изменяя соот-

ветствующие параметры виртуального процесса. Программа-модель на этапе отладки работает в цикле с программой регулирования. UltraLogic имеет библиотеку моделей объектов, способных достаточно точно моделировать поведение теплосистем.

Наконец программа загружена в контроллер и, конечно, в ней что-то надо подправить. Программисту совсем не обязательно одиноко сидеть в темном подвале рядом с контроллером. А если контроллеров несколько и установлены они в разных зданиях?.. UltraLogic содержит средства удаленной отладки и осциллографирования.

В режиме удаленной отладки UltraLogic имеет связь со всеми контроллерами, включенными в сеть. Кроме этого, программу можно отлаживать с нескольких компьютеров, имеющих статус Master. Отладчик запрашивает у контроллеров список переменных для визуализации и выводит их значение в специальном окне или в нескольких окнах. Значения переменных можно «на ходу» изменять (например, при подборе коэффициентов регуляторов), а также осциллографировать в виде графиков в соответствующих окнах.

Заключение

В приведенном проекте системы жизнеобеспечения используются контроллеры, имеющие открытую архитектуру РС. В качестве вычислительных модулей в контроллерах используются процессоры фирмы Octagon Systems (США). Контроллеры имеют системную шину, куда могут быть установлены любые модули ISA из номенклатуры MicroPC.

Пользоваться или нет специальными инструментальными системами программирования — дело вкуса и материальных возможностей конкретных творческих коллективов. Конечно, приятно тешить себя мыслью, что знаешь ассемблер или Си, и с их помощью вроде бы все можно сделать (возвели же люди голыми руками Великую Китайскую стену). Разработка подобных систем АСУ ТП еще 2-3 года назад могла составить честь специальному КБ с приличной численностью выдающихся тружеников интеллектуального труда. Сегодня с помощью UltraLogic на разработку проекта автоматизированной системы жизнеобеспечения в среднем уходит 1-1,5 месяца работы одного программиста. ●