

Принципы организации ПО в коммуникационных системах

Данная статья является заключительной в серии статей о коммуникационном контроллере ЭЛСИ-КОМ [1], [2]. В предыдущей статье «Связующее звено в технической системе» было дано краткое описание аппаратной организации контроллера, в этой статье будет рассказано о программной реализации всех функций коммуникационного контроллера, преимуществах и недостатках данного подхода, предлагаемой архитектуры.

Павел **Нестеренко**,
компания ЭлеСи



Программное обеспечение контроллера включает в себя программное обеспечение центрального процессора и программное обеспечение модулей. Структура ПО контроллера представлена на рис. 1.

ПО центрального процессора содержит операционную систему ElesyX и исполняющую систему ElesyTMS. Операционная система ElesyX представляет собой специально адаптированную и оптимизированную для аппаратного обеспечения контроллера ЭЛСИ КОМ версию ОС Linux. Исполняющая система ElesyTMS представляет собой набор исполняемых файлов и сценариев. Она автоматически запускается операционной системой при включении контроллера, осуществляет поддержку модулей контроллера и регулирует передачу данных между ними. Важным компонентом исполняющей системы является конфигурация контроллера — структурированный набор файлов, содержащий в себе информацию о составе контроллера, параметрах работы отдельных модулей и описания потоков данных между модулями.

Основная особенность построения программного обеспечения коммуникационного контроллера — модульный подход не только к аппаратной, но и программной части. Взаимодей-

ствие между модулями осуществляется посредством передачи сигналов. В состав контроллера входят аппаратные модули — коммуникационные устройства, обеспечивающие поддержку необходимого интерфейса связи. Также в состав контроллера входят программные модули. Они не являются физическими устройствами, но, так же, как и аппаратные, осуществляют прием, передачу и обработку данных. Примерами программных

модулей служат модуль логической обработки OpenPCS runtime, модуль ModBus TCP Client и т.п. На рис. 2 представлена функциональная схема взаимодействия модулей в системе.

Каждый модуль, входящий в состав конкретной конфигурации контроллера, характеризуется порядковым номером. Для аппаратных модулей порядковый номер совпадает с номером позиции на коммутационной панели. В контроллере предусмотрена одно-

временная поддержка до 16 различных модулей. Таким образом, порядковый номер модуля может иметь значение от 0 до 15.

В таблице 2 (журнал «itech», №3, стр. 38) описаны доступные на текущий момент модули. Список данных модулей постоянно расширяется в связи с поддержкой новых протоколов, добавлением новых функциональных возможностей.

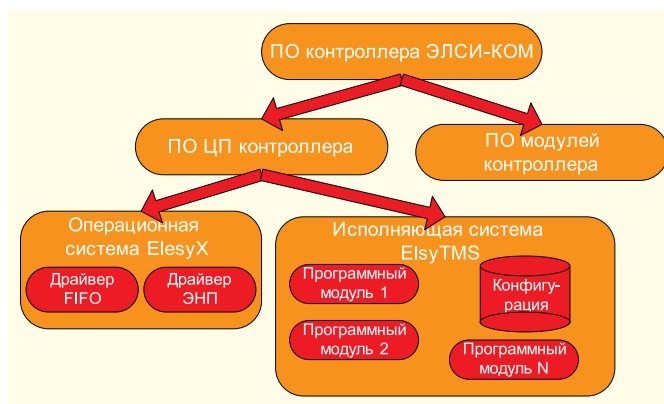


Рис. 1. Структура ПО контроллера

Одна из наиболее актуальных проблем при организации каналов обмена информацией – обеспечение надежного функционирования системы. Одно из решений – резервирование каналов связи. По поводу резервирования можно написать отдельную статью, для контроллера ЭЛСИ-КОМ проблема резервирования решается тремя разными способами.

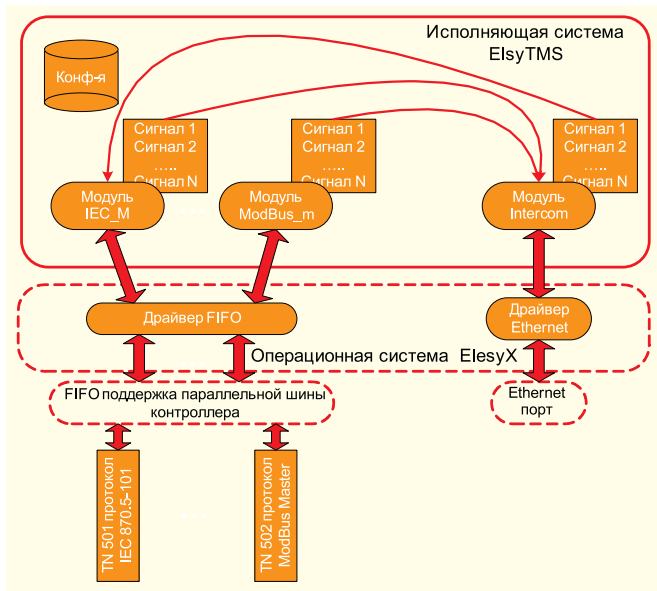


Рис. 2. Функциональная схема взаимодействия модулей в системе

1. Резервирование каналов связи на уровне аппаратного модуля. В частности, в состав контроллера входит модуль TN 503, имеющий два независимых интерфейса связи. Резервирование каналов связи осуществляется еще на уровне коммуникационного модуля, исполняющая система контроллера об этом даже не знает.
2. Резервирование однотипных протоколов связи на уровне программного модуля. Основная сложность в таком подходе – сложность программного обеспечения, получается связать только однотипные по функциональности протоколы связи.
3. Резервирование любых протоколов связи на уровне задаваемой логики обработки сигналов. В состав ЭЛСИ-КОМ входит программный модуль OpenPCS [3], который позволяет связать различные протоколы и интерфейсы связи.

На рис. 3 показана функциональная схема резервирования однотипных каналов связи и разнотипных каналов связи.

В контроллере ЭЛСИ-КОМ **сигнал** – основная единица данных, которыми оперирует контроллер. Сигнал имеет значение, тип, статус и идентификатор (адрес). Набор свойств приведен на рис. 4.

Каждый модуль контроллера рассматривается как поставщик и потребитель некоторого набора сигналов. Модули поставляют в контроллер сигналы, соответствующие аналоговым или дискретным входам, элементам диагностической информации или результатам логической обработки данных.

Модули могут принимать сигналы от других модулей и передавать в другие модули. Для хранения данных и обмена данными между модулями в контроллере существует база сигналов. База сигналов представляет собой набор сигналов, сгруппиро-

ванных определенным образом (см. рис. 2). Сигналы модулей контроллера, которые записываются в базу сигналов модуля, автоматически рассылаются одному или нескольким потребителям, определенным в конфигурации.

Каждый хранящийся в базе сигнал имеет набор свойств. Перечень свойств сигнала и их описание приведены на рис. 4.

Адрес сигнала – число, идентифицирующее поставляемый в базу сигналов модуля сигнал, уникальное в пределах конкретного модуля.

Адрес может состоять из нескольких компонент, структура адреса зависит от конкретного типа модуля. Существует правило символического описания адреса, используемое при описании сигналов в файлах конфигурации. Адрес описывается строкой вида (квадратные скобки означают, что заключенные в них компоненты необязательны).

Этот способ описания адреса используется, как правило, при описании сигналов коммуникационных модулей (МЭК, Modbus). Компоненты адреса в этом случае являются элементами адресации, используемой в коммуникационном протоколе. Их количество и диапазон допустимых значений зависят от конкретного типа модуля.

Тип сигнала задает способ хранения значения сигнала в контроллере. Тип имеет символическое представление, исполь-

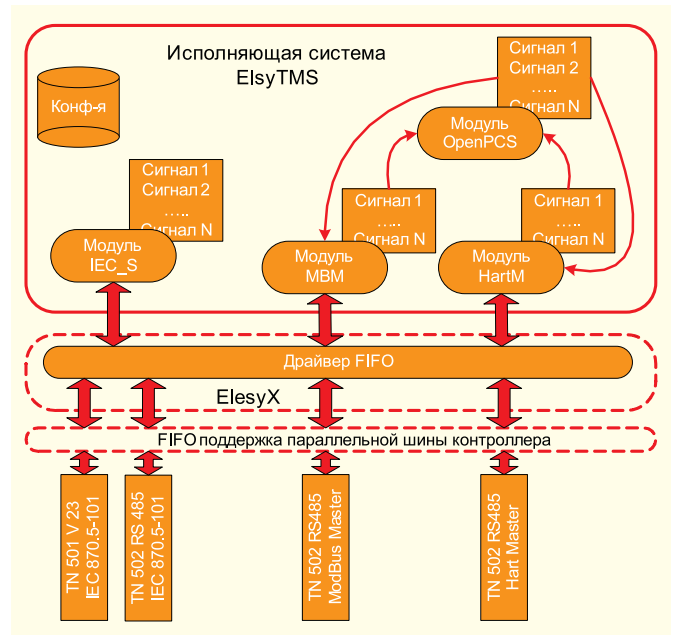


Рис. 3. Функциональная схема резервирования однотипных и разнотипных каналов связи

уемое при описании сигналов в файлах конфигурации. Типы сигналов приведены на рис. 4.

Значение сигнала – записанное модулем значение.

Статус сигнала – набор дополнительных характеристик сигнала. Он представляет собой набор 16 бит, формируемый модулем – источником сигнала и исполняющей системой.

Метка времени сигнала определяет время возникновения сигнала. В некоторых случаях, например, при работе с аппаратными модулями, метка времени устанавливается исполняющей системой в момент записи сигнала в базу. При-

← принципы организации ПО в коммуникационных системах



Рис. 4. Свойства сигнала и их описание

ный из модуля сигнал может также содержать свою метку времени. Время возникновения сигнала фиксируется с точностью до 1 мкс.

Тип доступа определяет возможность изменения сигнала с помощью диагностических средств.

Маршруты сигнала определяют рассылку сигнала потребителям. При записи сигнала в базу происходит автоматическая рассылка данного сигнала потребителям с присвоением ему нового адреса сигнала, индивидуального для каждого потребителя. Маршруты представляют собой список таких адресов.

Описание сигнала – краткий текстовый комментарий, описывающий сигнал. Применяется в файлах конфигурации.

ПАРАМЕТРЫ МОДУЛЕЙ

Каждый модуль контроллера имеет набор параметров, используемый для передачи в модуль при его инициализации, а также для корректной программной поддержки модуля исполняющей системой.

Существует возможность редактирования параметров модулей в работающем контроллере с использованием пульта инженера.

Взаимодействие элементов ПО контроллера во время работы и маршрутизация сигналов схематично показаны на рис. 5.

Каждый сигнал в контроллере может иметь одного поставщика и нескольких потребителей. Под поставщиками и потребителями понимаются аппаратные и программные модули.

Процесс конфигурирования контроллера заключается в создании или редактировании файлов конфигурации,

содержащих информацию о составе контроллера, параметры работы модулей и параметры маршрутизации данных. Конфигурирование производится с помощью программы ElsyTManager. Попробуем решить реальную задачу перераспределения потоков информации на простом примере.

ПРИМЕР ЗАДАЧИ КОНФИГУРИРОВАНИЯ:

Допустим, в нашей системе управления существуют три независимые подсистемы (очень реальная ситуация), все три системы используют для работы разные интерфейсы и протоколы связи. На рис. 6 представлена схема распределения информации.

Кроме преобразования одного протокола в другой, задача усложняется необходимостью нормирования (логической обработки части сигналов) собираемой информации из подсистемы 2. Для этой цели применяется программный модуль OpenPCS (нужно задать для него необходимый алгоритм работы на одном из языков программирования стандарта IEC61131-3).

Первый шаг – определение состава модулей коммуникационного контроллера (рис. 7).

Второй шаг – определение параметров работы модулей. Основные и расширенные параметры работы. Например, для модуля ModBus RTU Master необходимо задать основные параметры работы и таблицу опроса. Это же необходимо сделать и для модуля ModBus TCP Client (рис. 8-9).

Третий шаг – определение маршрутов (правил преобразования протокольных адресов) между модулями. Для этого необходимо выбрать в модуле-поставщике сигналов один или более сигналов и, следуя указаниям конфигурирования, установить

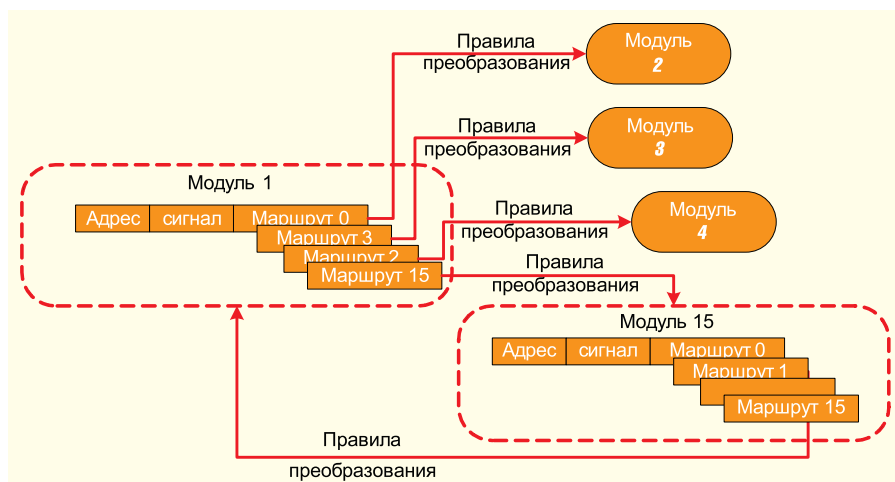


Рис. 5. Взаимодействие элементов ПО и маршрутизация сигналов



Рис. 6. Схема распределения информации

маршруты. Конфигуратор при этом попросит указать модуль, в который будут маршрутизировать выбранные сигналы, и правила преобразования адресов сигналов. На рис. 10 показан процесс маршрутизации сигналов модуля ModBusRTU Master в модуль IEC Slave.

Для логической обработки сигналов в конфигурацию добавлен модуль OpenPCS, правила работы с этим модулем ничем не отличаются от остальных модулей, основные принципы работы модуля OpenPCS приведены в [3]. Прделав аналогичные шаги для каждого модуля в системе, мы получим законченное решение – коммуникационный контроллер.

Как видно из примера, конфигурирование коммуникационного контроллера - задача довольно простая. Естественно, в реальной жизни объекты автоматизации гораздо сложнее приведенного примера. Описанный пример был сделан всего за несколько минут и содержит не более 100 сигналов. В реальных проектах количество сигналов доходит до 10000, и разработка проекта занимает несколько дней, а иногда и недель. Но даже на таких больших задачах процесс конфигурирования коммуникационного контроллера - задача не очень сложная. Заложенные в ПО контроллера принципы помогают максимально оптимизировать процесс конфигурирования. 🛠

СПИСОК ЛИТЕРАТУРЫ

1. П.Г. Нестеренко, М.Н. Гончарова И.В. Волженин. «Проблемы информационного обмена в современных системах автоматизации». Журнал «itech - интеллектуальные технологии», октябрь 2005 г.
2. П.Г. Нестеренко, А.В. Климов «Связующее звено в технической системе». Журнал «itech - интеллектуальные технологии», февраль 2006 г.
3. В.А. Агафонов «OpenPCS – работаем с «железом»!». Журнал «itech - интеллектуальные технологии», февраль 2006 г.

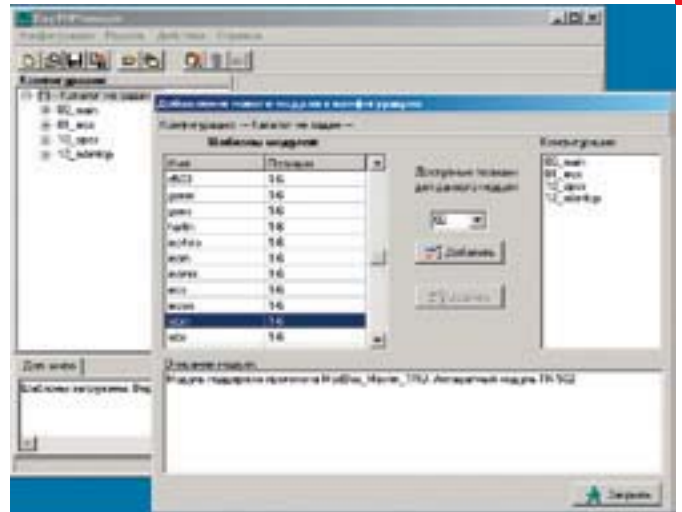


Рис. 7. Определение состава модулей

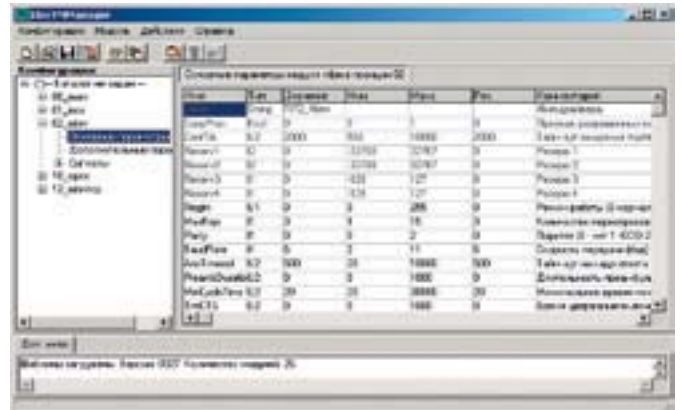


Рис. 8. Определение параметров работы

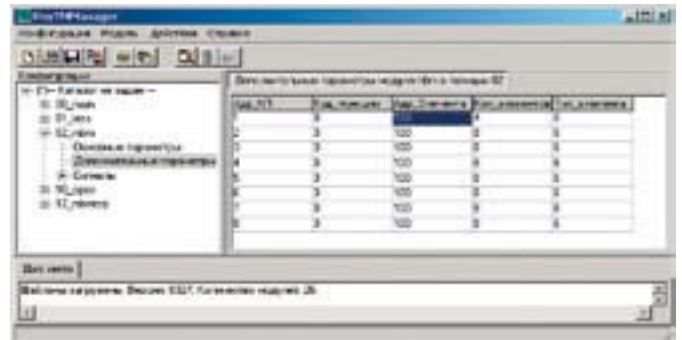


Рис. 9. Определение параметров работы

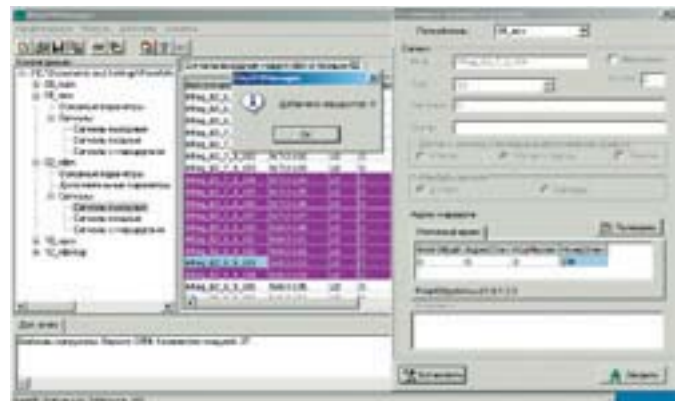


Рис. 10. Процесс маршрутизации сигналов