

НОВЫЕ ТЕХНОЛОГИИ В МИРЕ ПРОГРАММИРУЕМЫХ КОНТРОЛЛЕРОВ

И.Г. Варламов (Компания ИнСАТ)

Рассмотрено использование новых информационных технологий, реализованных в современных SCADA-системах, для программирования контроллеров на примере MasterSCADA 4D.

Ключевые слова: SCADA-система, визуализация, ПЛК, кроссплатформенность, Internet-технологии, двухслойная структура проекта.

Немного истории

Бурное развитие вычислительной техники во второй половине XX века привело к широкому применению специализированных микропроцессорных устройств — контроллеров, предназначенных для управления различными технологическими процессами. Одновременно начали формироваться различные подходы к работе с контроллерами.

Одно из направлений — свободно конфигурируемые контроллеры. Как правило, это небольшие контроллеры узкого назначения с ограниченным набором функций, использование которых задается в простом интерфейсе, зачастую в табличном виде, хотя в некоторых случаях возможно даже создание логических схем или процедур. Преимуществом такого подхода стала простота конфигурирования, не требующая привлечения специалистов высокого уровня со знанием программирования. Недостатком является весьма ограниченная область применения.

Другое направление — свободно программируемые контроллеры. Это наиболее распространенный тип контроллеров, практически не имеющий ограничений в области применения. Здесь следует разделять два подхода к программированию — «классическое» программирование на языках Basic, Pascal, C, C++ и им подобных и программирование на специальных языках, приближенных к технологическим. Появление специальных языков программирования было продиктовано несколькими обстоятельствами. В первую очередь сыграла роль сама область применения контроллеров. Еще на заре своего появления новые микропроцессорные устройства стали приходить на замену релейных или электронных схем управления, зачастую размещались в тех же шкафах и использовали те же внешние подключения. Тогда и появились первые языки, понятные инженерам, обслуживавшим прежнее оборудование. Например, язык релейной логики LD позволял буквально нарисовать электрическую схему, загрузить ее в контроллер, который продолжал выполнять те же функции, что и замененное оборудование. Появились и другие графические языки программирования, значительно упрощающие отладку и дальнейшее сопровождение программ.

Со временем был создан стандарт МЭК 61131-3, регламентирующий языки программирования контроллеров. На сегодняшний день в стандарт входят: язык релейной логики LD, язык функциональных блоков FBD, язык диаграмм функциональных последовательностей SFC и язык структурированного

текста ST. Языки этого стандарта стали основой для программирования ПЛК и их поддерживали во всех популярных средствах программирования контроллеров [1, 2].

Новые веяния

Сегодня новая волна революционных информационных технологий дала возможность значительно расширить назначение контроллера как логического управляющего автомата. Благодаря развитию Internet-технологий и унификации систем отображения информации уже никого не удивляет возможность управлять системой отопления дачного дома или сигнализацией автомобиля с мобильного телефона. Эти же технологии вошли и в сферу автоматизации и диспетчеризации более крупных объектов, где контроллеры стали выполнять функции, не присущие им ранее, традиционно осуществляемые сопряженными системами. Например, задача визуализации и организации взаимодействия человека с системой управления еще недавно была прерогативой SCADA. Сегодня необходимость собственной системы визуализации является обычным требованием практически к любой среде программирования контроллеров. Но не только наличие системы визуализации в полной мере отвечает современным потребностям к средствам разработки. Высокая степень децентрализации, широкая поддержка Internet-технологий, возможность работы в гетерогенных сетях — все это стало неотъемлемой частью современных систем автоматизации. Инструментальная среда, позволяющая создавать такие системы, уже не просто средство программирования контроллеров — это среда, объединяющая функции разработки SCADA, Softlogic, Web-приложений и др. Одним из таких продуктов является вертикально интегрированная платформа для разработки систем управления — MasterSCADA 4D.

Рассмотрим на ее примере, каким образом и за счет каких информационных технологий обеспечиваются требуемые качества.

Кроссплатформенность

Кроссплатформенность необходима не просто ради поддержки различных операционных систем. Благодаря кроссплатформенности появляется другое важное качество — унификация исполнения любых компонентов проекта на устройствах различного типа. Кроссплатформенность исполни-

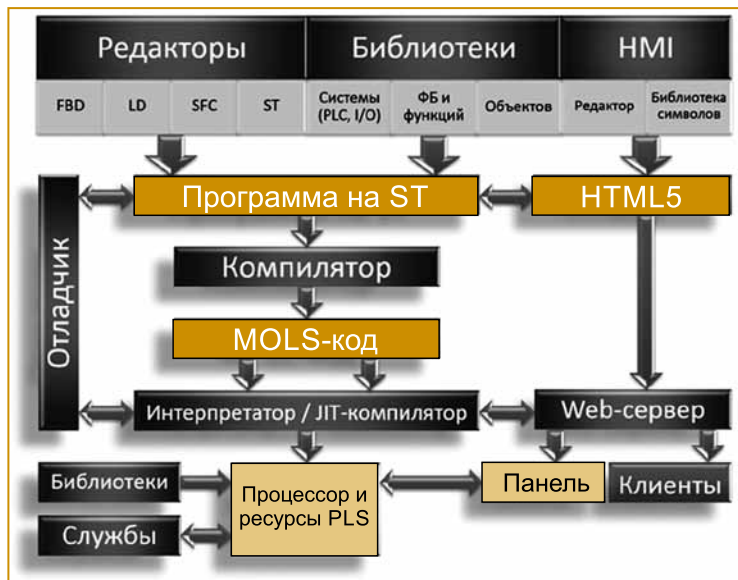


Рис. 1. Структура исполнительной системы MasterSCADA 4D

тельной системы достигается за счет применения интерпретируемой среды исполнения либо за счет применения JIT-компиляторов. Разумеется, сама исполнительная система в таком случае уникальна для каждой платформы и должна быть предварительно портирована на каждый тип устройств. Так, например, MasterSCADA 4D имеет исполнительную систему, работающую на ОС Windows, Linux, Android, QNX, Эльбрус и Unix (рис. 1). [3]

Интеграция различных устройств и гетерогенность

Другое важное свойство — возможность полной вертикальной и горизонтальной интеграции различных устройств распределенной системы в единый проект, использующий единое информационное пространство. Это означает, что в рамках одного проекта, разработчику не нужно специально связывать различные устрой-

ства какими-либо протоколами и создавать каналы связи межконтроллерного обмена или обмена с верхним уровнем. При компиляции проекта все необходимые связи будут созданы автоматически. А поскольку связи между узлами основаны на платформонезависимом коммуникационном стеке OPC UA, это дает возможность работы системы в гетерогенных сетях, состоящих из устройств различного типа и работающих под управлением разных ОС.

Свободная миграция функциональности

Предыдущие два свойства в совокупности предоставляют возможность свободной переносимости функциональности с одного устройства на другое. В MasterSCADA 4D любой функциональный блок, любой компонент проекта можно назначить для исполнения на том или ином устройстве через контекстное меню. Никаких дополнительных операций для этого не потребуется. Это позволяет гибко перераспределять различные компоненты проекта, а следовательно, и нагрузку в зависимости от имеющейся вычислительной мощности и легче адаптировать программно-логическую часть проекта для исполнения в системах различной топологии и состава.

Кроме того, это свойство позволяет использовать языки стандарта МЭК 61131-3 не только для программирования логики контроллеров, но и для любых других компонентов системы, включая компоненты визуализации, в том числе исполняемые непосредственно в графическом клиенте.

Двухслойная структура проекта

В классическом подходе иерархия проекта строится, исходя из его физической структуры. Например, имеется контроллер с определенным набором входов/выходов, и именно он находится на вершине иерархии. Далее в контроллер добавляется программа, работающая именно с этими входами/выходами, а также другие компоненты проекта. На первый взгляд такая структура понятна и легко ассоциируется с реальным физическим миром. Однако вопрос перераспределения ресурсов при таком подходе становится затруднительным, а в некоторых случаях и полностью невозможен. Удалили из проекта контроллер — исчезло и все его содержимое.

Суть двухслойной структуры заключается в том, что вся логическая часть проекта строится полностью автономно и независимо от физической структуры системы и входящих в нее устройств. Более того, при разработке логической структуры проекта, физической может еще и не быть. После создания логической структуры проекта она привязывается к физической структуре — переменные проекта связываются с конкретными входами/выходами или внешними каналами, программы и мнемосхемы назначаются для исполнения в те или иные устройства.

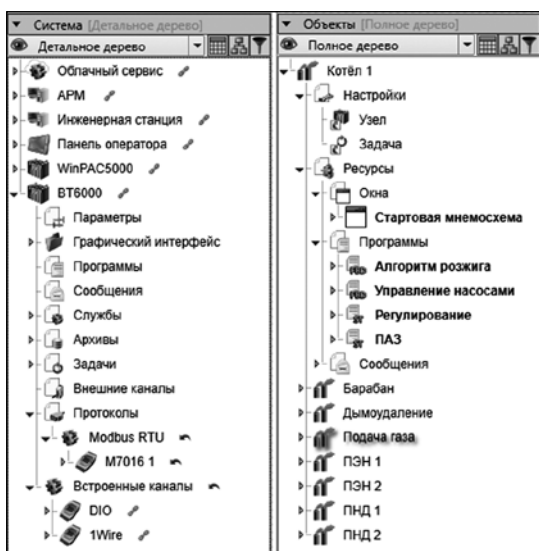


Рис. 2. Двухслойная структура проекта: слева дерево физической структуры, справа – логической

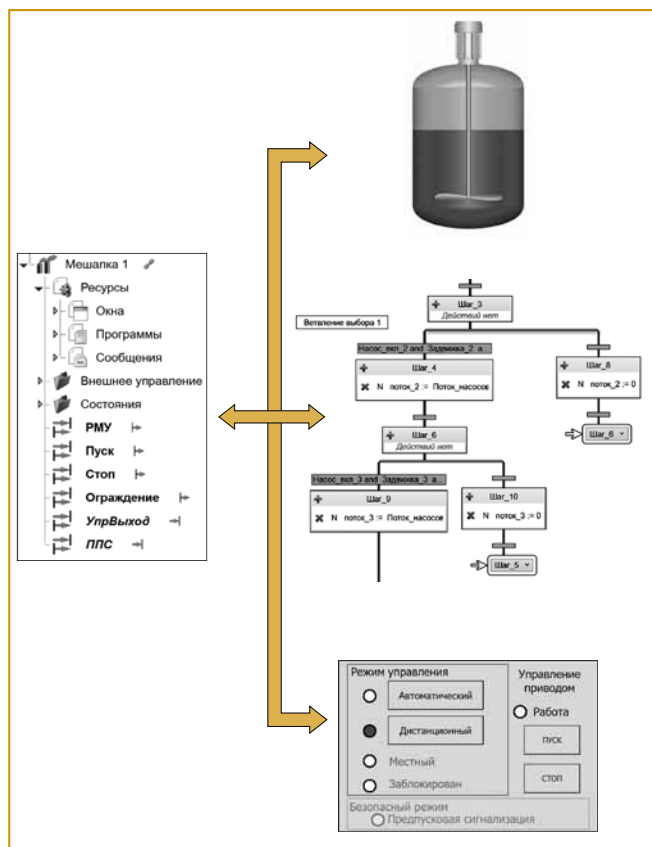


Рис. 3. Структура объекта в MasterSCADA 4D

Двухслойная структура проекта (рис. 2) хорошо знакомая пользователям MasterSCADA 3 [4].

Объектный подход

Логическая структура проекта основана на объектном подходе. Объект представляет собой именованную совокупность разнородных компонентов проекта, таких как программы, данные, мнемосхемы, журналы, сообщения и другие, в том числе вложенные объекты, а также связи между этими компонентами. Связь с внешними по отношению к объекту элементами проекта осуществляется через набор параметров объекта, представляющих интерфейс этого объекта — своего рода аналог клеммников. Объект как виртуальная логическая структура чаще всего ассоциируется со своим технологическим прототипом. Основное назначение такой компоновки — упрощение процедуры тиражирования компонентов части проекта, отвечающих за управление типовой технологической установкой. Объект копируется с сохранением всей внутренней структуры, включая внутренние связи между компонентами, и разработчику остается лишь соединить внешние (рис. 3).

Система визуализации

Унификация системы визуализации и организации человеко-машинного интерфейса в целом основана на стандарте HTML5, что позволяет использовать



Рис. 4. Унификация отображения на различных устройствах

как собственный графический клиент, так и штатные браузеры на планшетах, мобильных телефонах и других устройствах (рис. 4). В MasterSCADA 4D исполнительная часть системы визуализации представляет собой кроссплатформенный Web-сервер, который может физически располагаться как вместе с исполнительной системой сервера данных, так и отдельно, на другом устройстве или в “облаке”. Обмен данными между сервером визуализации и сервером данных осуществляется посредством Web-сервисов.

Заключение

Современные средства программирования контроллеров эволюционировали далеко за пределы своего первоначального предназначения и по своей функциональности представляют собой вертикально интегрированные платформы для разработки систем управления. Унификация исполнительной системы и средств визуализации на устройствах различного типа делают системы более гибкими, а проектные решения легко переносимыми и менее зависимыми от конкретной аппаратной реализации. Это существенно упрощает повторное использование разработчиками уже имеющихся решений, сокращает сроки разработки проектов и их стоимость.

В то же время поддержка средствами разработки новых информационных технологий расширяет и функциональность самих контроллеров, наделяя их качествами, не присущими ранее. Одним из примеров таких инновационных платформ является MasterSCADA 4D, разрабатываемая компанией ИнСАТ.

Список литературы

1. Петров И.В. CoDeSys - повседневный инструмент программиста ПЛК // Автоматизация в промышленности. 2012. №8
2. Колтунов А.В., Золотарев С.В. ISaGRAF6.1: динамичное развитие и концептуальные новшества // Автоматизация в промышленности. 2012. №8
3. Варламов И.Г. SCADA нового поколения. Эволюция технологий – революция системостроения // Автоматизированные информационно-управляющие системы в энергетике. 2016. №2 (79).
4. Аблин И.Е. Бритва Оккама – в надежных руках // Промышленные АСУ и контроллеры 2008. № 8.

Варламов Илья Геннадьевич – ведущий специалист компании ИнСАТ.
 Контактный телефон 7(495) 989-22-49.
 E-mail: info@insat.ru [Http://www.insat.ru](http://www.insat.ru)