



СОВРЕМЕННЫЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ — НОВОЕ КАЧЕСТВО ЖИЗНИ КОНТРОЛЛЕРОВ

ИЛЬЯ ВАРЛАМОВ
info@insat.ru

Веб-доступ к различным устройствам и сервисам, облачные технологии, «Интернет вещей» — все это стало неотъемлемой частью повседневной жизни. ИТ не обошли стороной и системы автоматизации: благодаря современным возможностям программирования контроллеры приобрели новые качества.

В настоящее время контроллер может быть самодостаточным звеном в небольших системах управления (уже никого не удивит возможностью управлять климатом в загородном доме или вести видеонаблюдение через Интернет). В то же время появилась возможность более тесно интегрировать контроллеры

в распределенную систему управления. За счет чего это произошло? Контроллеры, с точки зрения их физической реализации, остались практически те же. Новые качества им придадут системы программирования нового поколения, характерными особенностями которых являются высокая степень децен-

трализации, унификация системы визуализации и исполнительных систем на устройствах разного типа, широкое применение интернет-технологий и облачных решений. Эти возможности размывают прежде считавшиеся неизбывными границы между верхним и нижним уровнем, а также в классификации

устройств по их функциональному назначению.

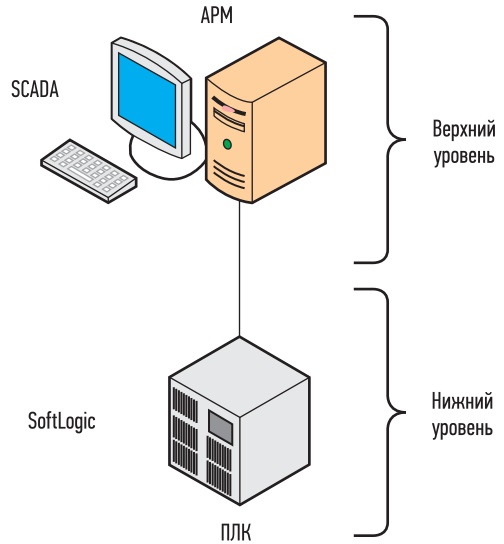
В «классической» структуре автоматизированного управления (рис. 1) у каждого устройства было строго свое функциональное назначение и для каждого из них существовали свои средства разработки. Контроллеры осуществляли опрос полевого оборудования, первичную обработку сигналов, вычисление логики управления и выработку управляющих сигналов. Они представляли собой нижний уровень систем управления, для программирования которого применялись, как правило, специализированные средства, получившие название SoftLogic. В распределенных системах, где требуется сбор данных от устройств нижнего уровня, отображение информации на экранах мониторов, ведение баз данных, формирование журналов событий, обеспечение аварийной сигнализации, генерирование отчетов, это осуществляется с помощью SCADA, реализованной на персональных компьютерах, которая представляет собой верхний уровень. Системы программирования бывают вертикально интегрированными, т. е. сочетают в одной среде разработки и SCADA, и средства программирования контроллеров — SoftLogic. В зависимости от сложности, системы построены по одиночной или по клиент-серверной архитектуре, но, независимо от этого, остается разделение на верхний и нижний уровни, каждый из которых функционирует на устройствах своего типа и программируется своими средствами.

Системы нового поколения, благодаря кроссплатформной среде исполнения и визуализации, предоставляют возможность свободно перемещать любые компоненты проекта между устройствами не только различного типа, но и построенными на различных программно-аппаратных платформах. Например, систему визуализации (ранее принадлежавшую АРМ) можно перенести в контроллер и получить доступ к нему с помощью тонких клиентов с других устройств сети. Это позволяет сделать контроллер самостоятельным узлом для небольших систем. И наоборот, часть функций контроллера с целью экономии

его ресурсов можно делегировать другим узлам, таким как виртуальный или облачный сервер. Одним из примеров системы программирования нового поколения является MasterSCADA 4D.

Исполнительная система MasterSCADA 4D является кроссплатформной и поддерживает такие операционные системы, как Windows, Linux, Android, QNX, Эльбрус и Unix не только на уровне контроллера, но и на всех уровнях систем управления — локальных HMI-панелях, АРМ оператора, серверах, облачных сервисах. Межузловое взаимодействие основано на стандарте OPC UA, что дает возможность построения систем в гетерогенных сетях и обеспечивает бесшовное (незаметное для разработчика) соединение между различными устройствами: любая переменная процесса, любой параметр одинаково доступны для любых компонентов проекта независимо от места их физического расположения. Средства программирования разработчикам SoftLogic языками стандарта МЭК 61131-3: ST (язык структурного текста), FBD (язык функциональных блоков), SFC (язык функциональных последовательностей) и LD (язык релейной логики). Благодаря унификации исполнительной системы эти языки теперь используются для программирования логики не только контроллеров, но и любых других компонентов системы, включая компоненты визуализации, в том числе исполняемые непосредственно в браузере.

Среда разработки реализована по принципу раздельного построения физической структуры проекта и его логической части — программной и графической (рис. 2). Такой подход позволяет легко переносить готовый проект на другую систему, имеющую не только различные аппаратные средства, но и иную структуру. Программно-логическая и визуальная часть проекта при этом остаются неизменными, требуется только переконфигурировать физическую структуру проекта — пере назначить узлы и оборудование. При разработке логической части проекта традиционно используется объектный подход, суть которого состоит в объединении некоторой



совокупности прикладных программ, данных, графических элементов и других элементов проекта, включая внутренние связи между ними, в отдельную именованную сущность. Это значительно упрощает работу с однотипными объектами.

Исполнительная система MasterSCADA 4D уже портирована на контроллеры многих отечественных и зарубежных производителей, среди которых ОВЕН, Ротек, ЭЛНА, TREI, MOXA, ICP DAS, Advantech и др.

Примером системы, реализованной на MasterSCADA 4D, в которой все логические компоненты проекта исполняются в контроллере, может

РИС. 1. ▲
«Классическая» система автоматизации одиночной архитектуры

РИС. 2. ▼
Дерево физической структуры проекта (слева) и дерево логической структуры (справа)

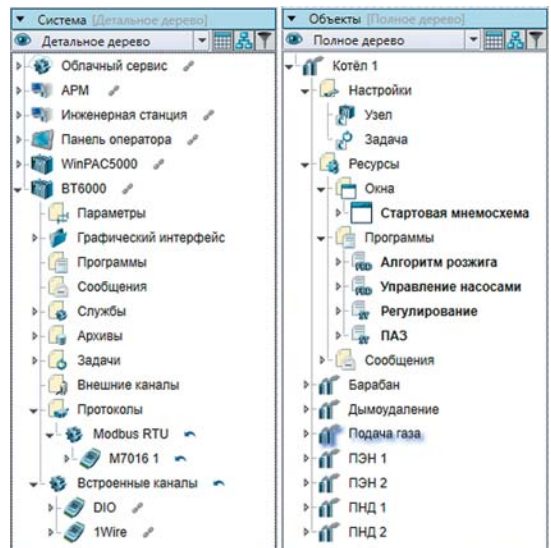


РИС. 3. ▶ Структурная схема системы диспетчеризации котельной

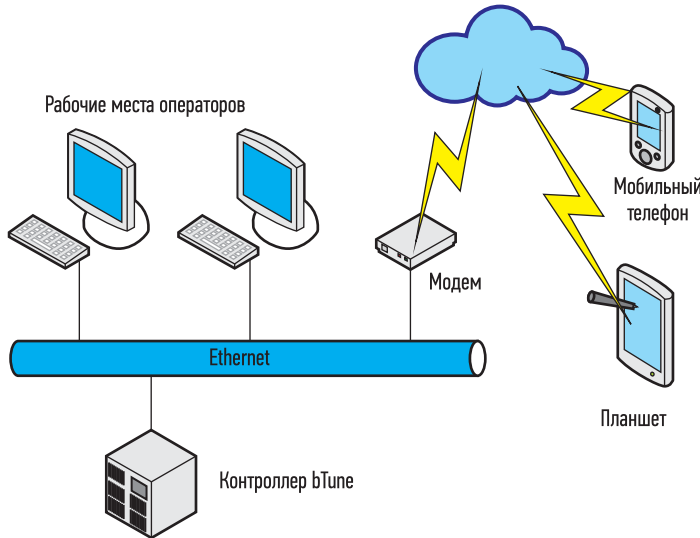
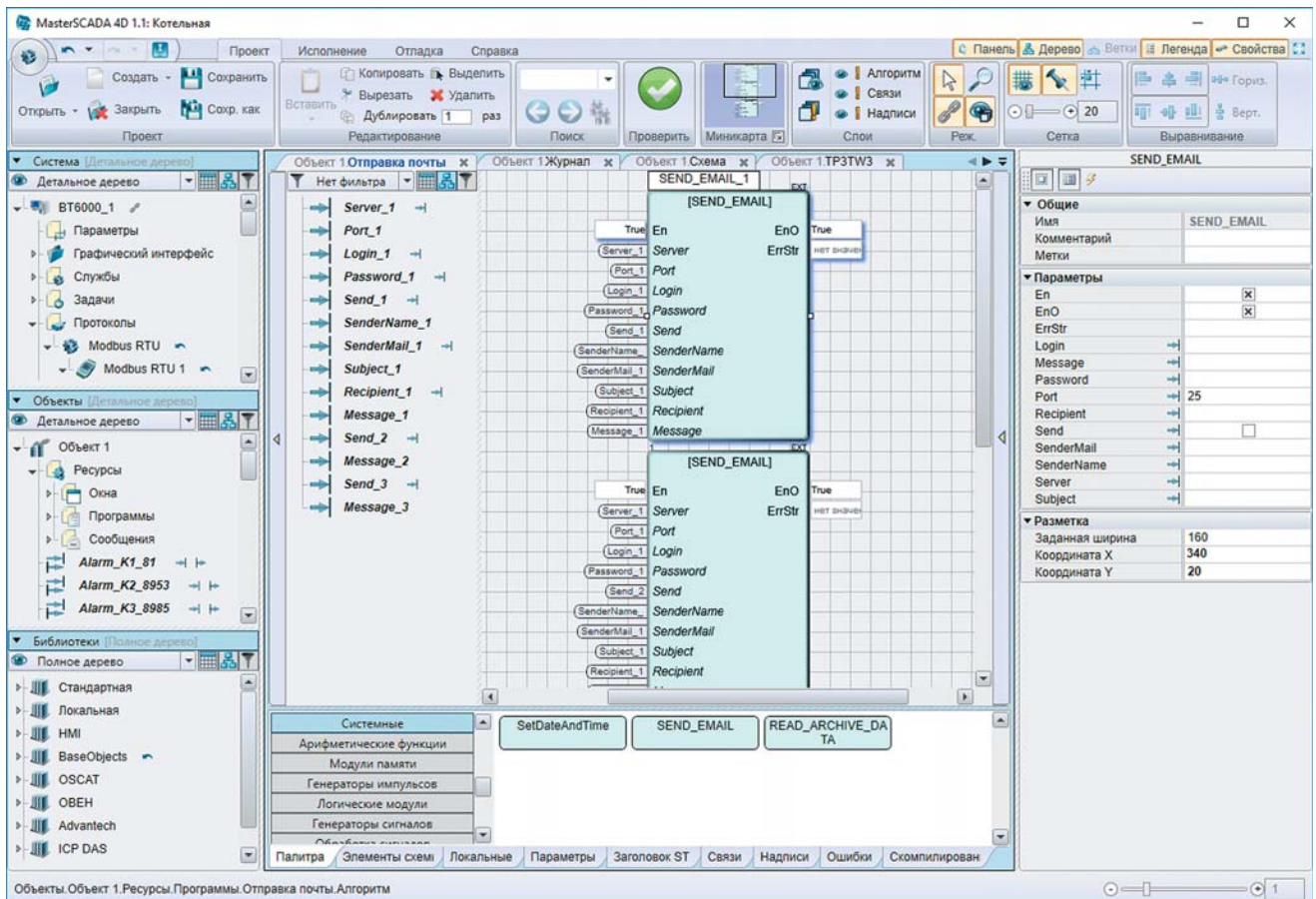


РИС. 4. ▼ Среда разработки MasterSCADA 4D. Проект диспетчеризации котельной



служить система диспетчеризации котельной — одного из объектов предприятия «Орловские тепловые сети» (рис. 3, 4). Система построена на базе контроллера bTune серии BT6000 производства компании «Ротек». Опрос периферийного и полевого оборудова-

ния производится по протоколу MODBUS. В контроллере реализована логическая часть проекта и система визуализации.

Логическая часть проекта производит опрос и обработку переменных процесса, их хранение

в краткосрочном архиве, выработку аварийных сообщений, отправку их по электронной почте и формирование журнала событий.

Система визуализации построена на базе веб-сервера, работающего в контроллере. Графическая часть интерфейса содержит мнемосхему (рис. 5), тренды переменных и журнал событий. Для доступа к веб-серверу используются штатные браузеры операторских станций, подключенных к контроллеру по сети Ethernet. К контроллеру также подключен ADSL-модем, благодаря чему одинаковый графический интерфейс доступен как на операторских станциях, так и через Интернет — с любых мобильных устройств, например планшетов или телефонов. Таким образом, в рассмотренной системе все необходимые функции реализованы непосредственно в контроллере и он является самостоятельным звеном.

Разумеется, исполнение всех функций в контроллере возможно лишь для относительно небольших систем. В случаях, когда требуется нака-

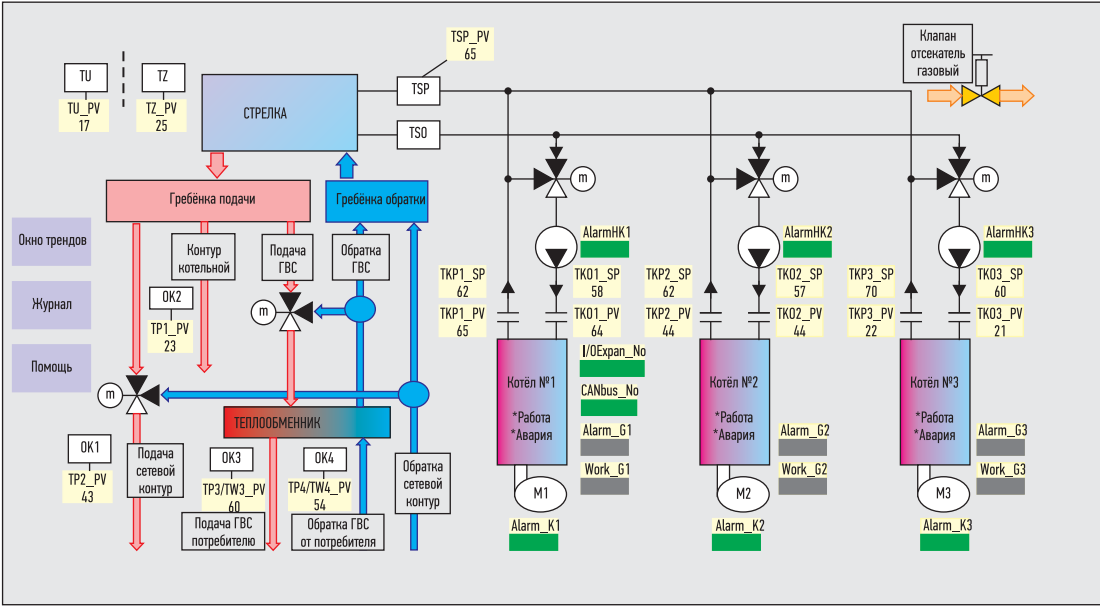


РИС. 5. Мнемосхема технологического процесса котельной

пливать долгосрочные архивы для большого количества переменных, интегрировать данные от нескольких устройств и иметь для них общую визуализацию, необходима более сложная архитектура системы [1]. Например, логика управления реализована в контроллере, а основные операции сбора, хранения и обработки данных производятся на сервере (рис. 6). Возможен также вариант переноса части функций (в том числе архивирования и визуализации) в облачный сервис. Тогда контроллер соединяется непосредственно с облачным сервером (рис. 7).

Важно отметить, что, независимо от архитектуры и сложности системы, все компоненты программируются в одной среде и одними и теми же средствами. Благодаря унификации исполнительной системы любой компонент, будь то программа, база данных или мнемосхема, одинаково выполняется как на контроллере, так и в облаке или на любом другом устройстве.

Развитие ИТ и современные средства программирования наделяют контроллеры новыми качествами, что предоставляет разработчику большую гибкость архитектурных решений в построении систем, а также дает возможность использовать одни и те же логические компоненты проекта для устройств различного типа и располагать

их на различных уровнях в иерархии систем управления. Благодаря этому расширяется спектр применения наработок, а сами решения становятся более унифицированными, что сокращает затраты на создание систем управления и делает труд инженера более эффективным и плодотворным. ●

ЛИТЕРАТУРА

1. Варламов И. Г. Средства автоматизации и вычислительной техники: импортозамещение как продолжение традиций // Control Engineering Россия. 2016. № 2.
2. Аблин И. Е. MasterPLC. Контроллер как открытая платформа // Автоматизация производства. 2011. № 2.
3. Аблин И. Е. Заметка о вкусном и здоровом контроллере // ИСУП. 2014. № 3.

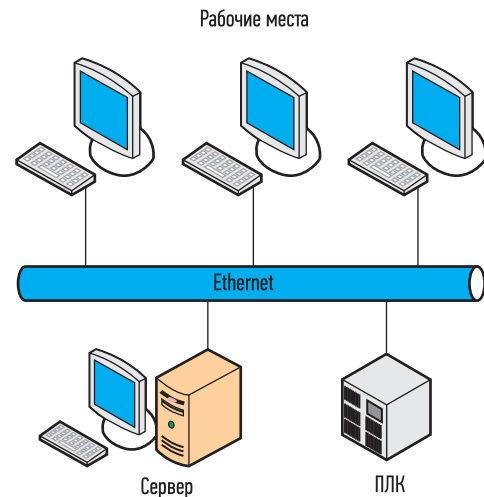


РИС. 6. Структурная схема системы на базе сервера

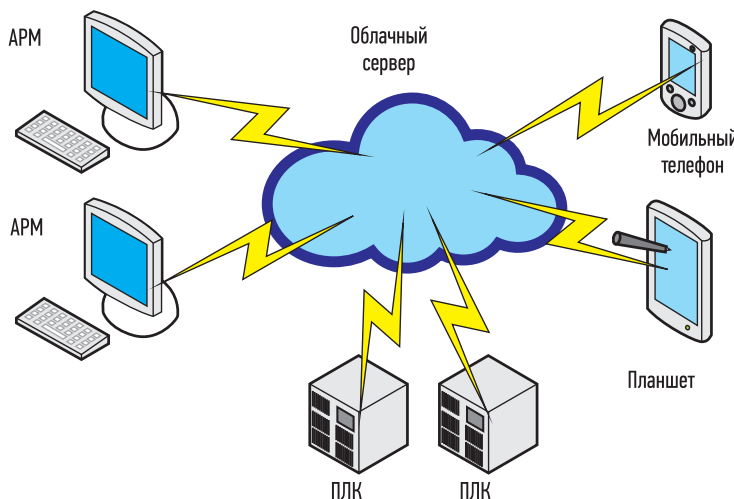


РИС. 7. Структурная схема системы на базе облачного сервера