

Средства пользовательского программирования и автоматизация проектирования в MasterSCADA



В статье рассматривается развитие средств пользовательского программирования в SCADA-системе MasterSCADA – от решения нестандартных задач управления и контроля на технологическом языке ST до автоматизации процесса проектирования во встроенной среде сценарного языка С#. В продолжение этой линии впервые анонсирована новая среда программирования контроллеров MasterPLC Designer, полностью реализующая требования стандарта МЭК 61131-3 и сохранившая принятую в MasterSCADA объектную идеологию, которая обеспечивает удобство и скорость разработки, тиражирование проектных решений.

ЗАО «НПФ ИнСАТ», г. Москва

От «перетаски и брось» к «напиши и запусти»

Объектно-ориентированная SCADA-система MasterSCADA изначально не содержала никаких средств программирования, даже традиционных сценарных языков (или на техническом жаргоне «скриптов»). Это объяснялось концептуальной позицией разработчиков, считавших необходимым приучать пользователей к объектной идеологии и стандартным инструментам MasterSCADA, обеспечивающих простым «перетаскиванием» (drag-and-drop) элементов проекта установление любых связей по передаче данных, а также включение одних элементов в другие (например, динамический символ или кнопку вызова документов одного объекта в мнемосхему другого). Тем не менее было необходимо найти возможности и для тех пользователей, которые хотели бы в рамках MasterSCADA решать нестандартные задачи.

Универсальный рецепт создания библиотечных блоков и визуальных контролов на языках профессионального программирования подходит далеко не всем инженерам. Еще одна причина озаботиться способами программирования прикладных задач была связана (начиная со 2-й версии MasterSCADA) с тем, что

MasterSCADA стала вертикально-интегрированной системой, в рамках которой можно было произвольно распределять логику контроля и управления между рабочими станциями и контроллерами с открытой архитектурой (для них была выпущена исполнительная система).

Первым инструментом технологического программирования в рамках MasterSCADA стал графический язык схем функциональных блоков. Но это – способ создавать решения, основанные на уже имеющихся библиотеках, а их функционал ограничен даже в условиях постоянного расширения. Решительное развитие необходимого инструментария началось уже в 3-й версии MasterSCADA. Библиотека функциональных блоков расширилась блоками пользовательских программ. Были реализованы блоки двух видов – для инженерного программирования на языке ST (стандарт МЭК 61131-3) и для автоматизации разработки проектов или реализации сложных задач на языке С#. Если программы на языке ST работают как на верхнем уровне систем, так и в контроллерах, то программы на С# предназначены исключительно для функционирования в рамках рабочих станций. Поскольку в MasterSCADA создается единый проект на всю систему у

автоматической организацией связи между ее частями, эту специализацию языков разработчик проекта должен учитывать изначально.

Программирование на языке ST

Язык ST прост для освоения инженерами. Поколениям, выпущенным из вузов в девяностых и нулевых годах, как правило, знаком из учебного курса язык Паскаль, от которого ST заимствовал основные идеи. К тому же программа на ST содержит чисто инженерные понятия – входы/выходы, переменные с типом «время» и т.п. Добавление в разделы INPUT или OUTPUT новых переменных автоматически приводит к появлению новых входов/выходов у функционального блока ST в проекте. Любые входы/выходы в проекте простым перетаскиванием могут быть связаны с входами/выходами других объектов или переменными проекта.

С точки зрения удобства разработки редактор ST предоставляет современную среду (рис. 1), встроенную, как это в MasterSCADA принято, непосредственно в интегрированный менеджер проекта так, что разработчик не задумывается ни над способом открытия редактора (просто выбирает закладку «Код» соответствующего блока в проекте), ни над тем, где хранятся файлы программы

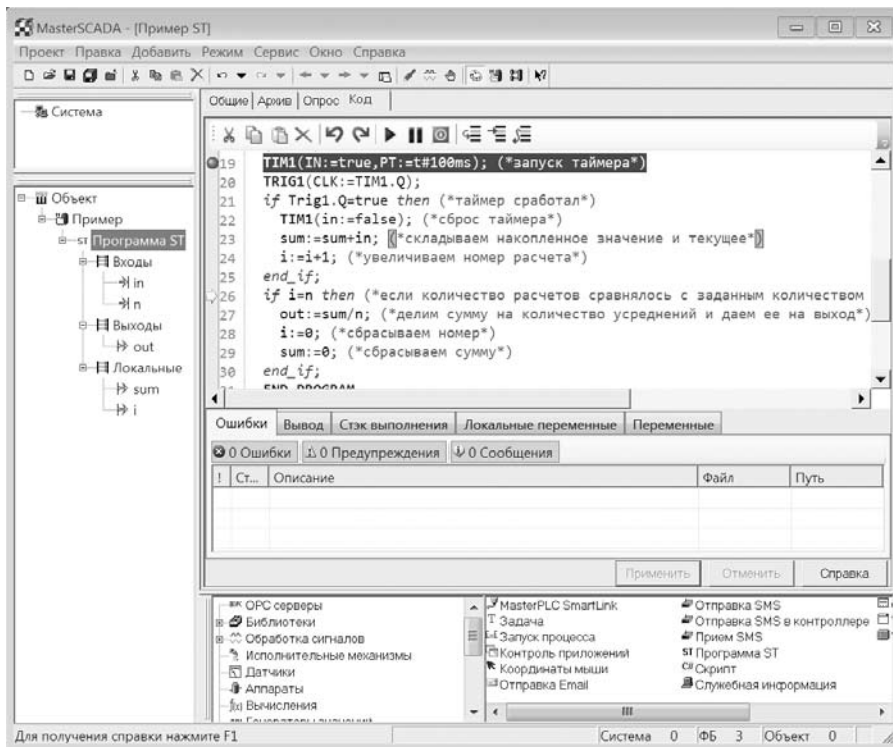


Рис. 1. Пример работы в редакторе ST, встроенном в MasterSCADA

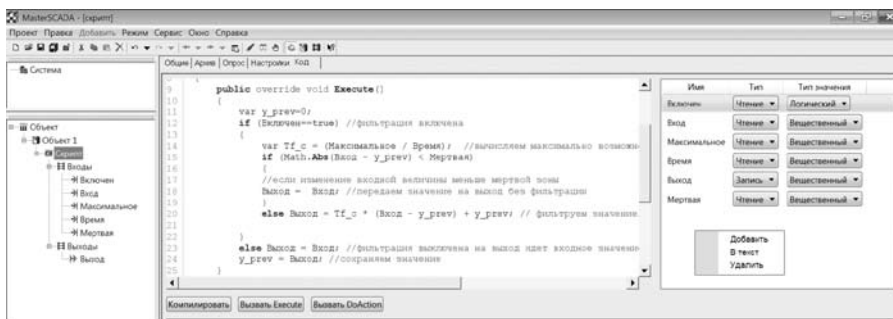


Рис. 2. Пример реализации вычислительного алгоритма на языке C#

(в MasterSCADA вопрос хранения и именования файлов решен системным образом и остается за кадром – разработчику не нужно знать, как называются и где хранятся эти файлы).

Созданная программа компилируется в специальный интерпретируемый код, который может выполняться и в компьютере, и во всех типах контроллеров, поддерживаемых входящей в состав MasterSCADA исполнительной системой MasterPLC. Это контроллеры с практически любыми современными (x86, ARM7, ARM9 и т.п.) и распространенными операционными системами (от DOSa и Windows CE до Linux и Ecos). Существенно и то, что для отладки не требуется иметь контроллер в наличии. Программу можно отлаживать как на так называемом Windows-контроллере

(исполнительной системе контроллера, запущенной на той же рабочей станции, что и проект MasterSCADA для верхнего уров-

ня), так и прямо в режиме разработки, запустив на исполнение код только одного разрабатываемого блока. При этом доступно традиционное исполнение программы по шагам, включая возможность входа во вложенные процедуры.

Программирование на языке C#

Для сравнения приведем пример программы на C#. В данном примере реализован вычислительный алгоритм фильтрации входного аналогового значения. Из рис. 2 можно увидеть, что переменные программы создаются на специальной панели заданием имени, выбором типа и разрешения на запись. Сразу после ввода они появляются в декларациях программы и входах/выходах алгоритма в дереве проекта.

Автоматизация проектов на языке C#

Задачи, с которыми сталкиваются в своей работе проектировщики, иногда требуют выполнения большого числа рутинных операций, а ведь передача компьютеру «механической» работы и есть главная цель автоматизации. Вот типичная задача – нужно создать систему поквартирного учета ресурсов в многоквартирном доме. Допустим, фрагмент проекта – учет в одной квартире – создан. Теперь его нужно размножить на требуемое число квартир и подъездов, создать мнемосхему с вызовом необходимой квартиры для просмотра ее показателей (рис. 3).

Именно для этих целей можно выполнять сценарии, написанные на

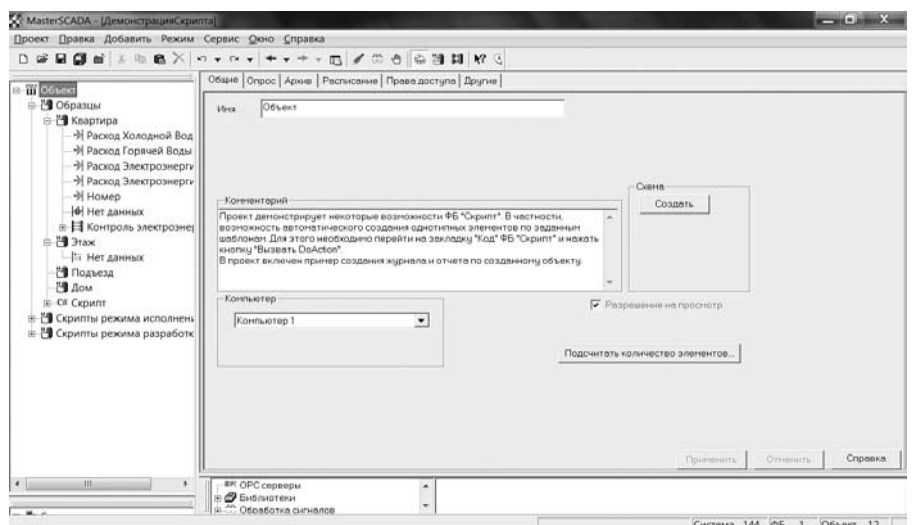


Рис. 3. Вид дерева проекта до выполнения сценария

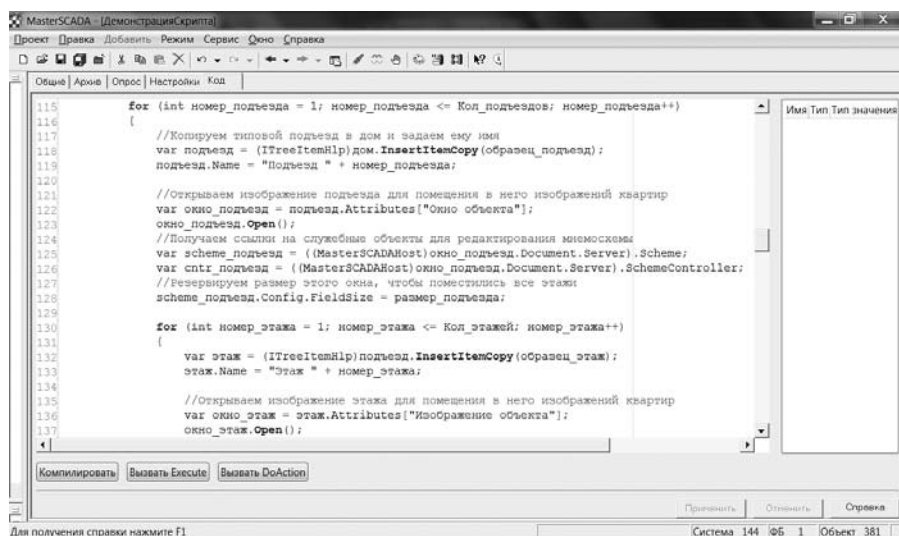


Рис. 4. Пример текста сценария, написанного на C# внутри MasterSCADA

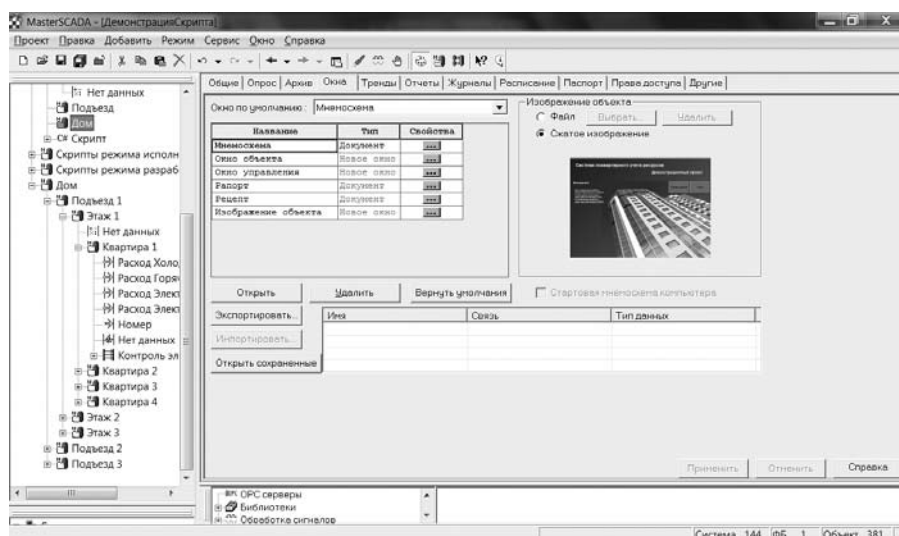


Рис. 5. Вид дерева проекта после выполнения сценария



Рис. 6. Вид мнемосхем с созданными сценарием кнопками вызова окон

C# внутри MasterSCADA и обращающиеся к ее объектной модели (рис. 4).

После запуска сценария (этот пример взят из библиотеки образцовых сценариев MasterSCADA) в проект добавляется новый объект

«Дом» на основе образца объекта «Дом». В него, исходя из заданных нами в настройках количественных параметров, вставляется указанное количество подъездов, этажей, квартир (рис. 5).

Кроме автоматического создания структуры проекта, сценарий создает мнемосхемы, журналы сообщений, отчеты и тренды, принадлежащие новым объектам, используя в качестве образцов документы исходного (созданного вручную) объекта. И, как окончательный результат, формируется главная мнемосхема проекта с автоматически созданными в необходимом количестве кнопками вызова окон квартир (рис. 6).

Подводя итоги рассмотрения этого примера, надо признаться, что поскольку MasterSCADA – очень большой программный продукт, то впервые сориентироваться в ее объектной модели, хотя она и документирована, сложно даже достаточно опытному разработчику. Поэтому услуги службы технической поддержки ИнСАТа по написанию сценариев автоматизации разработки проекта достаточно востребованы, особенно учитывая, что в большинстве случаев они бесплатны даже для пользователей демонстрационной версии, а созданные сценарии не пропадут даром, поскольку попадают в общую копилку – доступную всем библиотеку.

Логика развития – полная поддержка стандарта МЭК 61131-3

Резкий рост интереса пользователей к программированию контроллеров с открытой архитектурой на технологических языках, описанных стандартом МЭК 61131-3, привел нас к мысли не просто реализовать в MasterSCADA полную поддержку стандарта, но и выпустить отдельный продукт для тех, кто программирует контроллеры для автономного применения, а не для использования в рамках вертикально-интегрированных систем. Так появился MasterPLC Designer. Это полнофункциональная, полностью отвечающая стандарту интегрированная среда разработки, которая сохранила принятую в MasterSCADA объектную идеологию, позволяющую повысить не только качество проектов автоматизации, но и производительность труда проектировщиков. В связи с использованием новейшей программной архитектуры функционал этой среды не включен в состав

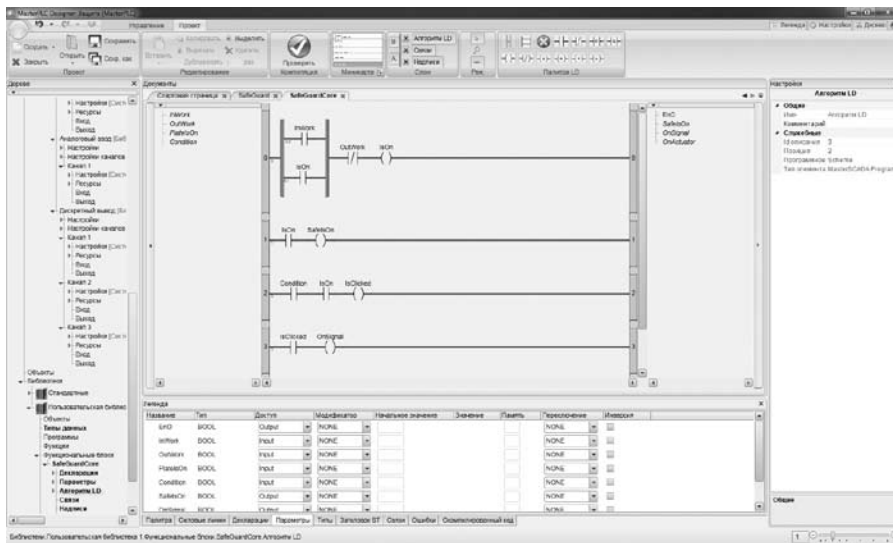


Рис. 7. Редактор схем релейной логики

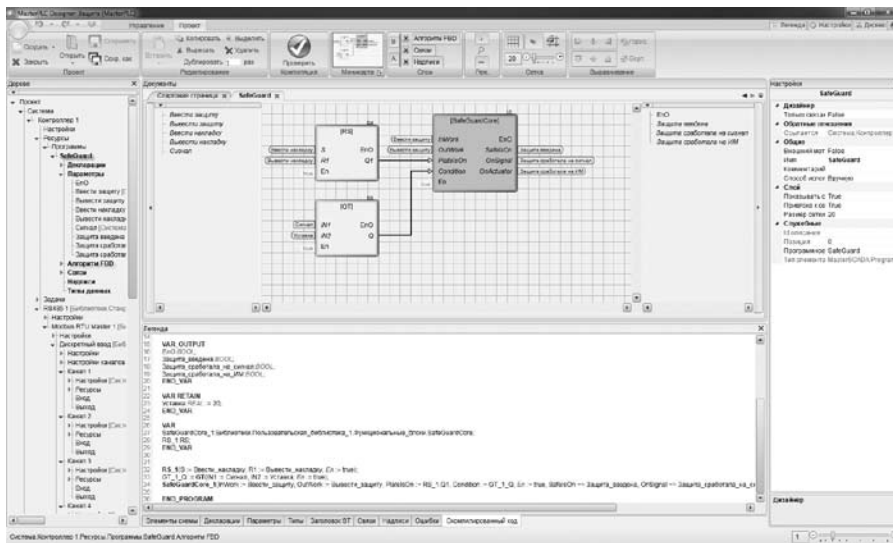


Рис. 8. Редактор схем функциональных блоков

создания программы одновременно на всех языках стандарта, используя для каждой части алгоритма тот язык, который для ее реализации окажется наиболее нагляден. Как правило, для динамических алгоритмов, например регулирования, удобнее использовать язык функциональных блоков (FBD), для описания логики инженерам-электрикам привычнее релейные схемы, а пооперационное управление прекрасно описывается языком шаговых последовательностей (SFC). Вычислительные и любые иные задачи могут быть решены на языке «Структурированный текст» (ST).

В приведенном на рис. 7 примере программа контроля параметра «нарисована» на языке LD и далее используется как библиотечный ФБ в программе, созданной уже на языке функциональных блоков (рис. 8).

Что дальше?

Особенность программной архитектуры MasterPLC Designer (и, следовательно, будущей MasterSCADA4, в состав которой этот продукт войдет) в том, что она полностью открыта для расширения. Это означает, что встроить в инструментарий поддержку еще одного графического языка (например, языка блок-схем для описания алгоритмов или языка UML для описания взаимодействия объектов проекта) мы сможем сразу, как только почувствуем его востребованность нашими потребителями. В случае MasterPLC Designer такими потребителями являются прежде всего производители контроллеров и их заказчики. Расширение списка поддерживаемых контроллеров происходит быстрыми темпами, и, следовательно, разнообразие запросов от растущей клиентской базы растет столь же быстро. Именно тесное взаимодействие с квалифицированными пользователями и является источником пополнения библиотек сценариев и алгоритмов, мотивом непрерывного развития средств разработки, способных справиться с усложняющимися задачами автоматизации.

MasterSCADA версии 3, но станет частью будущей 4-й версии.

Таким образом, сейчас в рамках инструментария MasterSCADA.3 ИнСАТ предлагает программировать системы, которые содержат как нижний контроллерный уровень, так и операторские станции, а проекты для автономных контроллеров рекомендуется разрабатывать с помощью намного более мощного нового продукта — MasterPLC Designer. Обе среды разработки используют одну и ту же исполнительную систему для контроллеров, поэтому с точки зрения технических характеристик и структурной функциональности (список под-

держиваемых контроллеров и платформ, драйверы, архивы, протоколы, быстродействие) их возможности практически одинаковы. Новую среду можно использовать и для программирования тех контроллеров, которые все же взаимодействуют с верхним уровнем, но используют для этого имеющийся OPC-сервер либо один из поддерживаемых протоколов, а не принятую в MasterSCADA технологию не требующего конфигурирования «прозрачного» обмена данных.

Посмотрим на пример проекта, разработанного в среде MasterPLC Designer (рис. 7). Основное преимущество новой среды — возможность

И.Е. Аблин, Генеральный директор,
 ЗАО «НПФ ИнСАТ»,
 тел.: (495) 989-2249,
 e-mail: ablin@insat.ru