

Программируемые логические контроллеры, МЭК системы программирования и CoDeSys

Любая автоматизированная система содержит в своем составе элементы, заставляющие ее выполнять нужные действия в нужном порядке, иными словами систему управления. Так для паровых машин был изобретен регулятор Уатта. С увеличением скорости вращения вала его подпружиненные шарики расходятся и через механическую тягу закрывают паровой клапан, уменьшая скорость вращения. Вторым достижением в сфере автоматизации по его исторической значимости можно назвать Жаккардовский ткацкий станок. Он позволял автоматически изготавливать полотна, составленные из нитей разного цвета по заданной программе. В последние десятилетия элементы управления систем автоматизации выполняются в виде электронных устройств. По сравнению с механическими системами они выигрывают по надежности, цене, габаритным размерам.

Возьмем для примера простейшую задачу: необходимо включить пресс через 1 секунду после одновременного удержания оператором двух кнопок в нажатом состоянии. Таким образом, мы гарантируем, что обе руки оператора заняты и даем ему время на контроль готовности машины. Самое простое решение это соединить контакты обеих кнопок последовательно и поставить электронное реле с таймером. Если таймер допускает регулировку времени задержки, то подобная схема обеспечит некоторую гибкость системы, впрочем не слишком высокую. Любые дополнительные условия, например требование контроля последовательности нажатий кнопок поставит нас в затруднительную ситуацию - мы будем вынуждены изменить схему, введя дополнительные реле. Это не является сложной проблемой при условии, что такая необходимость возникает крайне редко. Но в условиях конкурентного производства время выхода нового продукта на рынок имеет решающее значение и поэтому когда речь идет о гибком автоматизированном производстве, переналадка оборудования должна выполняться быстро, с минимальными затратами.

Дополнительной проблемой является увеличение сложности системы управления по мере развития производства и появления дополнительных функций (усложнения алгоритма работы). Любой специалист по автоматизации сталкивался также с проблемой построения системы управления для оборудования в той предметной области, которая ему недостаточно знакома: отсутствие четкой постановки задачи, появление новых условий по мере внедрения оборудования может сделать невозможной успешную реализацию проекта. Необходимо было создать управляющее устройство, алгоритм работы которого можно было бы менять, не переделывая монтажную схему системы управления, и в результате возникла логичная идея заменить системы управления с «жесткой» логикой работы (совокупность реле, регуляторов, таймеров и т.д.) на автоматы с программно заданной логикой работы. Так родились программируемые логические контроллеры (ПЛК). Впервые ПЛК были применены в США для автоматизации конвейерного сборочного производства в автомобильной промышленности (1969 г.).

Поскольку в определении «программируемый логический контроллер» главным являлось «программируемый», то практически сразу возник вопрос, как программировать ПЛК? Алгоритмические языки программирования компьютеров того времени были ориентированы на решение вычислительных задач. Профессия программиста считалась исключительно редкой и трудной, таких специалистов не было ни на одном производстве. Идеальным вариантом могла бы стать автоматическая трансляция принципиальных схем релейных автоматов в программы для ПЛК. Почему бы и нет? Так в ПЛК появился язык релейно-контактных схем (РКС или LD в английских источниках Ladder Diagram). Специалист-технолог мог «перерисовать» схему управления на дисплее программирующей станции ПЛК. Естественно схема изображалась не графически а посредством условных символов.

Например, описанная выше задача могла бы быть запрограммирована так:

кнопка1 кнопка2 таймер 1с выход1

|-----| |-----| |-----[T1]-----()---|

Слева и справа в такой программе мы видим вертикальные шины питания, соединенные горизонтальными цепями. Цепи могут состоять их контактов и некоторых дополнительных элементов (например, таймер) соединенных параллельно или последовательно. Справа каждая цепь заканчивается обмоткой реле. Контакты этого реле могут в свою очередь присутствовать в других цепях. Таким образом, можно составить достаточно сложную схему аналогичную по функциональности реальной релейной схеме. Первые программирующие станции представляли собой весьма громоздкие устройства, транспортируемые силами нескольких человек. Тем не менее, ПЛК активно начали заменять еще более громоздкие и главное обладающие “жесткой” логикой шкафы релейной автоматики.

Физически ПЛК представляет собой один или несколько блоков, имеющих определенный набор выходов и входов, для подключения датчиков и исполнительных механизмов (См. рис.1). Логика его работы описывается программно и выполняется встроенным микропроцессором. В результате, абсолютно одинаковые ПЛК могут выполнять совершенно разные функции. Для изменения алгоритма работы не требуется каких либо переделок аппаратной части.

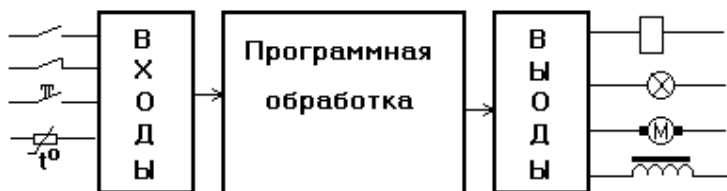


Рис. 1. Принцип работы ПЛК

Развитие электроники привело к потрясающей миниатюризации ПЛК. Сегодня существуют миниатюрные программируемые контроллеры, оснащенные небольшим дисплеем и встроенными возможностями программирования, подобные контроллеры получили название программируемых реле. Типовые задачи программируемых реле - очень простые локальные системы, имеющие до десятка входов и несколько силовых релейных выходов. Написать более сложную программу с помощью встроенного пульта не просто. Аналогично мы легко можем набрать текст SMS на клавиатуре сотового телефона, но даже ввод нескольких страниц текста, не говоря уже о больших объемах, представляется проблематичным. Для этого существуют персональные компьютеры (PC), предоставляющие гораздо более комфортабельные условия работы человека. Один современный ПЛК способен заменить десятки регуляторов, сотни таймеров и тысячи реле. Используя PC запрограммировать такую систему совсем не сложно. Применение PC в качестве программирующей станции ПЛК является сегодня доминирующим решением. Это не только упрощает программирование, но и решает задачи архивирования проектов, подготовки документации, визуализации и моделирования. Компьютер дает удобный универсальный инструмент как для программирования простейших локальных задач на ПЛК, так и для АСУ ТП.

Обратите внимание что, говоря о программировании ПЛК, мы все время возвращаемся к тому, как сделать этот процесс простым и удобным для человека. Казалось бы, однажды запрограммированный ПЛК будет работать годами и не очень важно будет ли его программа выглядеть красиво, главное чтобы она хорошо работала. К сожалению, это не так. Необходимость изменения программы в ПЛК возникает регулярно иногда и непредвиденно. Поэтому, написана она должна быть так, чтобы любой человек, а не только ее автор мог в ней быстро разобраться и оперативно внести необходимые доработки. Говорить о том, что программы написаны для ПЛК, не вполне корректно. Все программы написаны человеком и предназначены для чтения человеком. Любые инструменты программирования дают в конечном итоге микропроцессору инструкции в его машинных кодах. Для него нет разницы, на каком языке написана программа. В компьютерной науке доказано, что все языки программирования, обладают одинаковой вычислительной мощностью равной машине Тьюринга, на которой основаны все современные микропроцессоры. То есть, говоря о различных характеристиках

языков программирования, мы имеем в виду их способность эффективно выражать мысли программиста.

Упомянутый выше язык LD был изобретен в США в период релейной автоматизации. В Европу мода на ПЛК пришла несколько позднее, когда релейные шкафы были уже успешно заменены на шкафы с логическими микросхемами. Поэтому возникла необходимость изобретения других языков программирования понятных новому поколению инженеров. Так в Германии появились языки простых текстовых инструкций напоминающих ассемблер (IL). Во Франции возникли графические языки функциональных блоковых диаграмм (FBD) и высокоуровневые диаграммы описания этапов и условий переходов (Графсет, современный SFC). Применялись также языки, используемые для программирования компьютеров (Pascal, Basic). В конце семидесятых годов сложилась крайне сложная ситуация. Каждый изготовитель ПЛК (в том числе и в СССР) разрабатывал собственный язык программирования, поэтому ПЛК разных производителей были программно несовместимы, кроме того существовала проблема аппаратной несовместимости. Замена ПЛК на продукт другого изготовителя превратилась в огромную проблему. Покупатель ПЛК был вынужден использовать изделия только одной фирмы либо тратить силы на изучение разных языков и средства на приобретение соответствующих инструментов.

В итоге в 1979 году в рамках Международной Электротехнической Комиссии (МЭК) была создана специальная группа технических экспертов по проблемам ПЛК. Ей была поставлена задача выработать стандартные требования к аппаратным средствам, программному обеспечению, правилам монтажа, тестирования, документирования и средствам связи ПЛК. В 1982 году был опубликован первый черновой вариант стандарта, который получил наименование МЭК 1131. Ввиду сложности получившегося документа, было решено разбить его на несколько частей, вопросам программирования посвящена третья часть стандарта “Языки программирования ПЛК”. Поскольку с 1997 года МЭК перешел на 5 цифровые обозначения, в настоящее время правильное наименование международной версии части стандарта посвященной языкам программирования ПЛК – МЭК 61131-3.

Рабочей группой МЭК было принято достаточно оригинальное решение. Из всего многообразия существовавших на момент разработки стандарта языков программирования ПЛК были выделены 5 языков, получивших наибольшее распространение. Спецификации языков были доработаны, так что стало возможным использовать в программах написанных на любом из этих языков стандартизованный набор элементов и типов данных. Такой подход МЭК не раз подвергался критике, но время доказало правильность этого решения. Реализация подобного подхода позволила привлечь к программированию одного и того же ПЛК специалистов различных областей знаний (и что особенно важно – различной квалификации): специалистов по релейной автоматике (и даже электриков), программирующих в LD, специалистов в области полупроводниковой схмотехники и автоматического регулирования для которых привычен язык FBD, программистов, имеющих опыт написания программ для компьютеров на языке ассемблера (ему соответствует язык IL для ПЛК), на языках высокого уровня (язык ST), даже далекие от программирования специалисты-технологи получили свой инструмент программирования – язык SFC. Хотя внедрение МЭК систем программирования и не позволило полностью отказаться от услуг профессиональных программистов (впрочем такая цель и не ставилась), но зато позволило снизить требования к квалификации и соответственно затраты на оплату труда программистов ПЛК.

Стандартизация языков позволила (по крайней мере, частично) решить проблему зависимости пользователя ПЛК от конкретного изготовителя. Все современные ПЛК оснащаются средствами МЭК 61131-3 программирования, что упрощает работу пользователям контроллеров (можно использовать ПЛК различных фирм без затрат на переучивание) и одновременно снимает ряд проблем для изготовителей ПЛК (можно использовать компоненты ПЛК других изготовителей). Следует отметить, что некоторые старейшие изготовители ПЛК до сих пор вынуждены поддерживать свои собственные языки (системы программирования), однако все они, в той ли иной форме, стремятся обеспечить поддержку МЭК 61131-3.

Стандарт существенно расширил возможности на рынке труда специалиста, занимающегося программированием ПЛК. Подобно тому как автомеханик, имеющий стандартный набор инструментов, может браться за ремонт любого узла (кроме нестандартных) машины любой фирмы, так и специалист, изучивший языки МЭК 61131-3 сможет разобраться с программой любого современного ПЛК. Это позволило уменьшить как зависимость фирмы от специалиста по программированию ПЛК, так и специалиста от фирмы.

Еще одним важным положительным результатом стандартизации языков явилась возможность специализации изготовителей ПЛК как на производстве аппаратных средств ПЛК, так и на производстве средств программирования ПЛК. Результаты такой специализации хорошо видны на примере индустрии персональных компьютеров: существуют компании выпускающие высококлассные аппаратные средства, они умеют делать это лучше других и не испытывают необходимости выпускать программное обеспечение, в то же время на рынке программных средств есть свои лидеры, вооруженные опытом и имеющимися у них технологиями. Благодаря стандартизации обеспечивается совместимость и в результате пользователь может свободно выбрать лучшие продукты как из аппаратных, так и из программных средств.

В области систем МЭК 61131-3 программирования существует явное разделение на системы разного уровня. Дело в том, что обязательных требований в стандарте немного и поэтому можно создать весьма упрощенный инструмент, поддерживающий ограниченный набор элементов и объявить о совместимости с МЭК 61131-3. Нередко компании, начинающие производство ПЛК, так и поступают, стараясь всячески сэкономить на программном обеспечении. В результате их продукты приобретают репутацию дешевых поделок, даже обладая хорошими «аппаратными» характеристиками. Как изготовителей программистского инструментария для PC, так и изготовителей профессиональных высококлассных инструментов МЭК 61131-3 программирования для ПЛК в мире единицы.

На сегодняшний день лидирующие позиции на рынке МЭК систем программирования занимает комплекс CoDeSys немецкой компании 3S-Smart Software Solutions GmbH. Его применяют 190 компаний во всем мире, большинство из этих компаний - ведущие изготовители оборудования и/или систем промышленной автоматизации. В России ПЛК с CoDeSys хорошо известны специалистам, диапазон продукции, выпускаемой под управлением этих ПЛК огромен: от сигарет, соков и жевательной резинки до немецких автомобилей, оборудования судов и элитных яхт.

Цель настоящей статьи – дать общее представление о месте CoDeSys в МЭК системах программирования ПЛК и наиболее важных для пользователя особенностях CoDeSys, поэтому обратим внимание только на ключевые моменты и дадим самое общее представление о среде программирования. Подробная документация по CoDeSys на русском языке бесплатно доступна для пользователей на сайте www.codesys.ru.

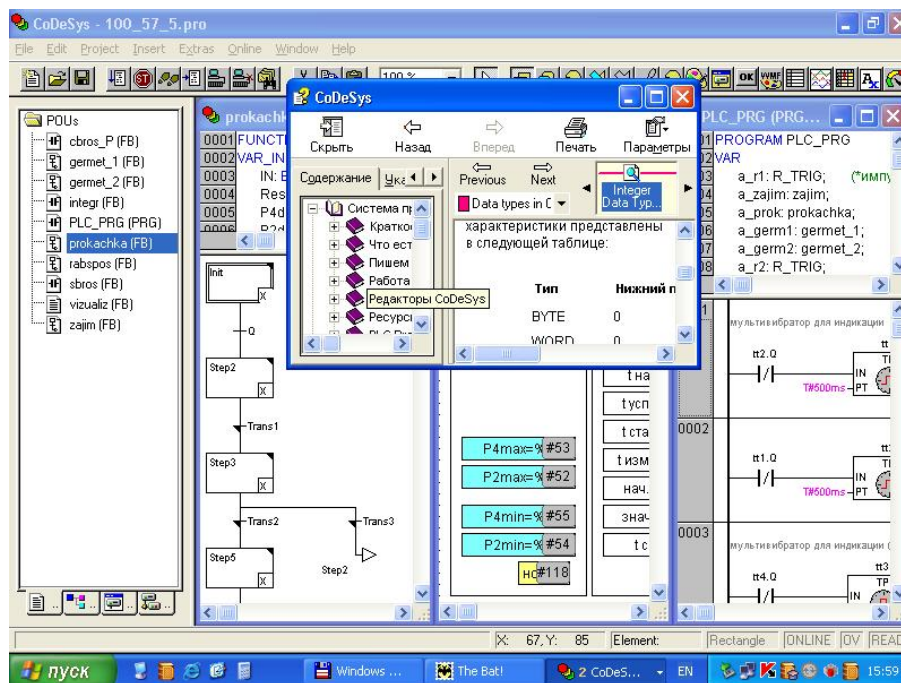


Рис. 2. Ввод программы в CoDeSys

CoDeSys включает 5 специализированных редакторов для каждого из стандартных языков программирования: Список Инструкций (IL), Функциональные блок-диаграммы (FBD), Релейно-контактные схемы (LD), Структурированный текст (ST), Последовательные функциональные схемы (SFC). Редакторы поддерживают большим числом вспомогательных инструментов, ускоряющих ввод программ (рис. 2). Это ассистент ввода, автоматическое объявление переменных, интеллектуальная коррекция ввода, цветовое выделение и синтаксический контроль при вводе, масштабирование, автоматическое размещение и соединение графических элементов [1].

В одном проекте можно совмещать программы, написанные на нескольких языках МЭК либо использовать один из них. Никаких особых требований по выбору языка нет. Он обусловлен исключительно личными предпочтениями. В России наиболее популярен язык ST. Это текстовый язык, представляющий собой несколько адаптированный Паскаль. Второе место по популярности занимает графический язык FBD, далее следует язык LD.

Помимо средств подготовки программ, CoDeSys включает встроенный отладчик, эмулятор, инструменты визуализации и управления проектом, конфигураторы ПЛК и сети. Методы программирования и отладки программ детально рассмотрены в цикле статей [2].

В основу CoDeSys положено несколько важных идей, которые выделяют его в группе лидирующих МЭК комплексов.

CoDeSys изначально задумывался как инструмент для профессионального применения и поэтому он не содержит каких-либо ограничений в реализации языков МЭК. Напротив, он включает ряд дополнений, не предусмотренных стандартом (языки CFC и упрощенный SFC, поддержка указателей и действий в функциональных блоках). CoDeSys компилирует прикладные программы в машинный код, поэтому создаваемые пользователями программы имеют наивысшее быстродействие. Комплекс CoDeSys активно развивается, в настоящее время он уже включает целый ряд расширений, таких как система контроля версий проекта (ENI), средства для создания приложений управления движением (SoftMotion), web-визуализация, библиотеки наиболее популярных функций, например, таких как регуляторы. Важным фактором здесь является то, что компания 3S молода и динамична и ее основатели [3] сами продолжают активнейшую работу над развитием CoDeSys, ведь очевидно, что плоды наемного труда и творческого вдохновения всегда будут отличаться. С первых лет своей работы компания 3S исключительно активно привлекает пользователей к творческому процессу. Компания ежегодно проводит конференции изготовителей и пользователей ПЛК, на которых обсуждаются не только текущие проблемы, но и оригинальные идеи развития комплекса. До 80% новшеств CoDeSys предложены именно пользователями. Ближайшая открытая конференция пользователей CoDeSys состоится 23 мая 2006 г. в России (Смоленск). В настоящее время идет активнейшая работа над принципиально новой версией комплекса CoDeSys. Она включает целый ряд передовых идей, включая поддержку объектно-ориентированного программирования на языках МЭК [3].

Использование CoDeSys предъявляет определенные требования к изготовителю ПЛК, связанные с реализацией одной из ключевых идей CoDeSys - переносом большей части проблем, возникающих при программировании контроллера, с пользователя ПЛК на изготовителя. В чем же проявляются эти требования? Обеспечение развитых сервисных функций требует наличия системы исполнения (специализированной операционной системы), размещенной в ПЛК, это означает, что пользователь не может взять любой ПЛК и запрограммировать его в CoDeSys. Предварительно изготовитель контроллера должен встроить в свой ПЛК специальным образом настроенную систему исполнения CoDeSys SP. Такая бизнес-модель требует определенных затрат и предварительного выполнения высококвалифицированной работы изготовителем ПЛК, принявшим решение абсолютной ориентации на пользователя ПЛК, который получает полностью готовый для работы контроллер и не должен ничего дополнительно приобретать или лицензировать. Поддержка CoDeSys, поэтому является верным признаком не только высокого технического уровня изготовителя ПЛК, но и выражает желание изготовителя обеспечить пользователю своих ПЛК максимально эффективные условия для работы. Идея поддержки пользователя ПЛК заложена и поддерживается разработчиком CoDeSys 3S - среда программирования CoDeSys (то есть та часть CoDeSys, с которой имеет дело пользователь) универсальна, принципиально не имеет никаких ограничений по числу установок, все обновления среды программирования можно загрузить с Интернет сайта компании 3S бесплатно.

Воплощением еще одной неожиданной идеи, коллективно сформированной пользователями CoDeSys, стало добровольное объединение изготовителей ПЛК, поддерживающих CoDeSys, в некоммерческую организацию CoDeSys Automation Alliance (CAA). Суть идеи в том, чтобы превратить изготовителей средств промышленной автоматизации, поддерживающих CoDeSys, в партнеров (насколько это возможно на конкурентном рынке) и нейтрализовать последствия конкуренции между изготовителями для пользователей ПЛК. Вместо намеренного создания технических препятствий, не позволяющих пользователям легко использовать продукты другой компании, члены САА целенаправленно принимают меры призванные обеспечить совместимость своих продуктов. Пользователь может быть уверен, что его прикладная CoDeSys-программа будет работать в любом контроллере любой компании являющейся членом САА. Пользователь может быть уверен, что используемые им инструменты (CoDeSys) проверены тысячами пользователей во всем мире. Пользователь всегда может обсудить свои затруднения и получить реальную помощь от широкого круга коллег, имевших опыт решения подобных задач. На сегодняшний день членами CoDeSys Automation Alliance являются 78 компаний. Участие в САА подчеркивает, что компания-изготовитель ПЛК обладает определенной бизнес-культурой и реальной уверенностью в своих силах выпускать продукты в соответствии с уровнем ведущих европейских фирм.

Литература

1. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования /Под ред. проф. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2004.
2. Петров И.В., Вагнер Р. Отладка прикладных ПЛК программ в CoDeSys (часть 3)// Промышленные АСУ и контроллеры. 2006. № 4.
3. Хесс Д., Объектно-ориентированные расширения МЭК 61131-3 // СТА. 2006. № 2.