

CODESYS ДЛЯ ПРОГРАММИРОВАНИЯ ВСТРАИВАЕМЫХ СИСТЕМ

ИГОРЬ ПЕТРОВ
ООО «ПК Пролог»



Во встраиваемых системах обычно нет необходимости в программировании пользователем. Соответственно, не должно быть нужды и в МЭК-языках. Большинство разработчиков встраиваемых систем слышали о CODESYS, но, как правило, относятся к нему с изрядным скепсисом. В их понимании МЭК-системы программирования применимы только для ПЛК. В статье предпринята попытка разобраться, справедливо ли такое мнение.

В системах промышленной автоматизации базовым интеллектуальным элементом является программируемый логический контроллер (ПЛК) [1, 2, 3, 6, 7]. С точки зрения программирования главная особенность ПЛК состоит в том, что для работы с ним не требуется образование в области информатики. Инструменты и языки программирования ПЛК должны быть максимально просты и в то же время эффективны. Упрощение использования не должно повлечь за собой ограничение круга решаемых задач.

Для удовлетворения этих противоречивых требований были созданы специальные языки программирования. В 1982 г. вышла первая редакция международного стандарта МЭК61131-3 (далее МЭК). В нем определено пять языков программирования ПЛК: три оригинальных визуальных и два, пришедших из мира компьютеров. Так, к первым относятся языки «релейных схем» (Ladder Diagram, LD), «функциональных блок-овых диаграмм» (Function Block Diagram, FBD) и «последовательных функ-

циональных схем» (Sequential Function Chart, SFC). А «Список инструкций» (Instruction List, IL) — это аппаратно-независимый ассемблер. Высокоуровневый язык «структурированный текст» (Structured Text, ST) представляет собой модифицированный Паскаль. Отдельные программные компоненты можно описывать на разных языках МЭК даже в одном проекте. В зависимости от уровня подготовки, типа решаемой задачи и личных предпочтений прикладной МЭК-программист выбирает оптимальный язык.

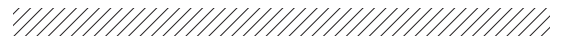
Создание качественного транслятора языка программирования высокого уровня является сложной и трудоемкой задачей. Для графических языков объем работы увеличивается за счет необходимости создания соответствующего графического редактора и отладчика. Поэтому задача поддержать несколько разных языков в одной программной среде стала серьезной проблемой для многих изготовителей ПЛК. В итоге это привело к возникновению компаний, специализирующихся на создании

универсальных сред программирования на языках МЭК. Одной из наиболее успешных оказалась немецкая 3S-Smart Software Solutions GmbH со своим комплексом CODESYS.

CODESYS включает в себя редакторы и трансляторы для всех пяти стандартных языков с рядом существенных расширений. Он также поддерживает значительное число специализированных отладочных и сервисных функций. На сегодня CODESYS — мировой лидер среди МЭК-комплексов. С его помощью ежегодно программируется более полумиллиона контроллеров. После долгих лет горячих споров 18.01.2013 г. была одобрена третья редакция стандарта МЭК 61131-3. В нее вошли оригинальные объектно-ориентированные расширения языков МЭК [4], впервые реализованные в комплексе CODESYS V3. Таким образом, CODESYS создал новый международный стандарт. Описание его составляющих, приемов эффективной работы и практике применения посвящено несколько книг и множество статей [1, 3, 6, 7, 8, 9].

ТАБЛИЦА. СРАВНИТЕЛЬНЫЙ АНАЛИЗ КОМПИЛЯТОРОВ C++ И CODESYS

C++	CODESYS
1. Трансляция кода/Среда программирования	
Компиляторы генерируют машинный код с качественной оптимизацией по размеру и быстродействию. Для разных семейств микропроцессоров обычно применяются разные компиляторы. Как правило, обновление версий компиляторов идет независимо от среды программирования.	Встроенные компиляторы генерируют машинный код с оптимизацией по надежности и переносимости. Тысячи разных устройств, на разных микропроцессорах, используют одинаковые компиляторы, что обеспечивает их качество. Обновление компиляторов идет вместе с обновлением среды программирования. Но есть опция выбора версии компилятора. Это позволяет исключить риск при правке старых проектов.
2. Интерфейс с системным уровнем (API)	
Функции чтения входов и записи выходов нужно писать в программе. На разных аппаратных платформах интерфейс с системным уровнем разный. Он меняется при замене процессора или ОС.	МЭК-программа работает с образом входов/выходов. Их обслуживает система исполнения. Доступ к специфическим устройствам выполняется через системные библиотеки.
3. Переносимость	
Многие функции прикладного кода могут быть использованы повторно. Системно-зависимые интерфейсы нужно адаптировать.	Если набор ресурсов (порты, полевые сети и др.) не изменился, то программа переносится без изменений. Необходимо переконфигурировать ввод/вывод и адаптировать системно-зависимые библиотеки.
4. Удобство прикладного программирования	
Естественная среда для компьютерных программистов, чего не скажешь о сервисном персонале. Модификации прикладного кода могут влиять на стабильности системы в целом. Должны выполняться только опытными программистами.	7 различных языков программирования, включая графические, позволяют программисту создавать код так, чтобы он был понятен прикладным специалистам и обслуживающему персоналу. Программист может выбирать оптимальный язык для разных функций.
5. Разделение между верхним/прикладным/пользовательским уровнем управляющей программы и системным уровнем	
Не реализовано. Для понимания работы управляющей программы нужно предварительно иметь представление об организации программного обеспечения в целом.	Правильно организованное МЭК-приложение имеет два или более уровня с четким разделением. Верхний уровень отображает общую структуру и алгоритмы управления. Это позволяет понимать работу системы обслуживающему персоналу, не имеющему специальной подготовки. Внутренние детали приложения скрыты (защищены) в функциональных блоках и библиотеках.
6. Объектно-ориентированное программирование (ООП)	
Было сильнейшим аргументом в пользу выбора C++ до выхода CODESYS V3.	Поддержано в CODESYS V3 для всех языков. Позволяет использовать современную технологию организации приложений.
7. Графический интерфейс оператора (HMI)	
Возможен. Требуется написания дополнительного кода или использования вспомогательных инструментов. Интерфейс отображения должен быть адаптирован для разных устройств вывода.	Встроенный графический редактор с набором типовых элементов отображения и ввода. Для управления графическим элементом нужно ввести наименования переменных программы в соответствующие поля. Визуализация отображается в среде программирования, на графической панели устройства или через Интернет. Способ отображения выбирается в диалоговом окне.
8. Отладка	
Среды программирования имеют встроенные отладчики. Дополнительные функции, типа графической трассировки, фиксации значений переменных, не практикуются.	Полная поддержка общепринятых отладочных функций (выполнение по шагам, условные точки останова, стек вызовов и т. д.) на всех языках МЭК. Дополнительные функции для отладки и обслуживания системы в целом (графическая трассировка, менеджер рецептов, фиксация переменных, визуализация). Возможно визуальное моделирование объекта управления.
9. Реальное время	
Определяется организацией программы. Зависит от ОС. Может вызвать ограничения при отладке.	Обеспечивается системой исполнения. В прикладной программе усилий не требует.
10. Поддержка стандартных полевых сетей (Fieldbus)	
Не включена. Требуется дополнительное ПО, реализующее стек протокола, и отдельный конфигуратор. Стеки аппаратно-зависимы. Конфигуратор (если есть) обычно не интегрирован в среду программирования. Символьное отображение входов/выходов проблематично.	Интегрированные конфигураторы и стеки полевых сетей. Стеки полевых сетей написаны в среде CODESYS и независимы от нижнего уровня. Нужный стек компилируется и линкуется с приложением автоматически.
11. Специализированный инструментарий для прикладных областей	
Нужно программировать самим или приобретать дополнительные инструменты. Интеграция обычно требует разделяемой памяти. Контроль движения и логика управляются разными процессами.	Интегрированный механизм сохраняемых переменных. Мониторинг входов и управление выходами без программирования. Специализированные прикладные библиотеки. Встроенная система управления движением (Soft Motion CAM/CNC), 3D-редактор перемещений, интерпретатор G-кодов (ЧПУ). Работает в одном процессе, единая адресация.
12. Командная работа/управление версиями	
Большинство сред разработки поддерживают контроль версий.	Поддерживается путем установки специального плагина для интеграции с Subversion.
13. Быстрое создание/тиражирование однотипных систем	
Существуют специальные приемы и инструменты, ускоряющие работу программиста.	Пакет CODESYS Application Composer позволяет конструировать приложения с автоматической генерацией надежного кода. Приложение составляется из готовых настраиваемых блоков без программирования.
14. Поддержка новых стандартов на рынке систем управления	
Нужно отслеживать и делать самостоятельно, заказывать или приобретать сторонние разработки.	Функционал, соответствующий новейшим мировым стандартам, наращивается постоянно. Например: FDT, OPC UA, EtherCAT, CANopen, PROFINET, ASI и т. д.
15. Контроль целостности/защита копирования	
Нет. Реализуется программно.	Встроенная защита модификации кода, уникальный идентификатор проекта. Опционально: шифрование проекта, многоуровневая система прав доступа, сохранение исходных текстов в памяти устройства.



АСПЕКТЫ ПРОГРАММИСТА

Чаще всего программисты встраиваемых систем противопоставляют CODESYS интегрированным компиляторам языка C/C++. Попробуем сравнить их технически (результаты сравнения приведены в таблице).

Как показано в таблице, для человека, имеющего образование по специальности программирование, C/C++ является естественным выбором. Переход к использованию МЭК-языков потребует некоторых усилий по освоению. Обычно начальный дискомфорт в CODESYS вызывает отсутствие главного цикла и функций ввода/вывода, которые полностью «спрятаны» в системе исполнения. Существенно отличается работа с таймерами. Ближе всего к языку C в CODESYS язык ST. Как правило, для его уверенного освоения программисту достаточно нескольких часов [5].

Использование МЭК-языков может не дать явных преимуществ мгновенно. Они проявляются ярко при необходимости пояснения прикладной программы другим людям. В этом смысле весьма эффективна связка языков SFC и ST. Диаграмма SFC визуально представляет интуи-

тивно понятный алгоритм работы, буквально «оживающий» в онлайн-режиме. Действия шагов SFC описываются на привычном высокоуровневом языке ST.

АСПЕКТЫ РУКОВОДИТЕЛЯ

Некоторые успешные практические применения CODESYS во встраиваемых системах иницировались не техническими специалистами, а менеджментом компании. Чаще всего главной причиной становилась одна из четырех, перечисленных ниже.

Проблема правильной организации работ

Как правило, в компании есть системный программист с соответствующим образованием. Он начинает с «оживления» платы, установки операционной системы (ОС) или самостоятельной разработки необходимого функционала нижнего уровня. Кажется логичным, чтобы он продолжил работу над системой далее, включая прикладные функции. Ему приходится потратить немало времени на то, чтобы разобраться с прикладной сферой. Будь то газовый котел, система управления яхты,

пульт авиационного тягача или что угодно, постепенно он начнет ориентироваться в теме не хуже заказчика. Иногда даже лучше, поскольку вынужден разбираться с причинно-следственными связями и выстроить алгоритмы. В итоге компания приобретает абсолютно уникального специалиста. Работа успешно сдается. Далее становится ясным, что только этот специалист способен сопровождать систему. Его нельзя отвлекать или заменять, нельзя предоставить отпуск или уволить. Компания попадает в зависимость от специалиста.

Вначале сам программист рад своей уникальной позиции. По прошествии нескольких лет он замечает, что «золотой фонтан» не забил, а работа стала утомительной. Он отстал от новых технологий по основной специальности. Сменить работу не реально. Проблема возникает с двух сторон.

CODESYS выручает благодаря свойствам 2 и 5 (см. таблицу). Нижним уровнем, включая установку системы исполнения CODESYS, занимаются системные программисты. Прикладной проект делается в CODESYS специалистами по прикладной области. Благодаря 4, им не требуется специальное образование. Системный и прикладной уровни четко разделены, как и требования к их исполнителям. При необходимости сопровождение ПО может быть передано заказчику. Специалистам заказчика достаточно пройти двухдневные учебные курсы.

Проблема развития универсальности системы и расширения рынка сбыта

Если устройство оказалось востребованным, то постепенно возникает необходимость специальных исполнений для разных заказчиков. Появляются несколько «прошивок» программы. Потом разрабатывается специальная компьютерная программа — конфигуратор системы. Она настраивает параметры устройства. Постепенно число параметров возрастает. Соответственно, разбухает документация. Но по-прежнему находятся заказчики, которым не хватает нужного именно им функционала. Рождается логичная мысль пойти дальше и дать заказчикам (дистрибьюторам, установщикам) возможности программирования системы. С языком C++ это нереально. CODESYS решает данную про-



▶
Эволюционный набор для PLC Core модулей SYSTEC с системой исполнения и визуализацией CODESYS V3

блему. Существует ряд специализированных контроллеров для печатных машин, деревообрабатывающих станков, заправочных станций, дорожных машин, химических аналитических приборов и др., которые оснащены CODESYS и дополнены специализированными библиотеками. Изготовитель соответствующей машины сам адаптирует контроллер под разнообразные модели и наращивает функционал. Изготовитель встраиваемой системы выигрывает за счет унификации и расширения областей применения.

В простых устройствах выручает встроенная визуализация CODESYS (п. 7 таблицы). С ее помощью графические конфигураторы, включая веб-интерфейс, делают «популярно» на ранних версиях прикладного проекта.

Проблема интеграции с устройствами других компаний

Обычно в успешно развивающейся системе возникает необходимость подключения приборов, датчиков и других устройств ведущих мировых изготовителей. Самостоятельная реализация стандартных полевых сетей значительна по трудоемкости. Покупка готового стека дорога. CODESYS (см. п. 10 таблицы) позволяет поддерживать практически все стандартные сети. В нашей практике к использованию CODESYS во встраиваемых системах чаще других приводила необходимость поддержки стеков CANopen и EtherCAT.

Проблема замены ПЛК встраиваемой системой

В некоторых случаях, помимо базового серийного изделия, требуются особые исполнения. Пример: компактный холодильник и заказная холодильная установка, различные термокамеры, весы, сварочные автоматы, научное оборудование и др. Под специальное исполнение целесообразно применить готовый ПЛК с CODESYS. На нашем рынке доступны десятки таких ПЛК разных ценовых категорий. В разных применениях могут использоваться компактные, модульные или панельные ПЛК. Для серийных изделий изготавливается собственный встраиваемый контроллер. Если везде стоит CODESYS, то одна команда справ-



ляется с разработкой ПО и сопровождением всех вариантов. Практически везде применяется один проект с разной конфигурацией.

УСТАНОВКА К CODESYS

Для того чтобы устройство программировалось в CODESYS, в нем предварительно должна быть установлена так называемая система исполнения CODESYS Control. Она включает планировщик задач, загрузчик, функции отладки, обслуживает полевые сети, ввод/вывод и т. д. Именно благодаря ей МЭК-программа оказывается аппаратно-независимой. Набор ресурсов, которые должна обслуживать система исполнения, отличается у разных контроллеров. Речь идет не только о микроконтроллере, но и об устройстве в целом. По этой причине нельзя просто скопировать систему исполнения с одного устройства на другое. Она всегда требует некоторой индивидуальной адаптации. Все существующие встраиваемые системы с CODESYS созданы одним из трех способов:

1. Бизнес-модель разработчиков CODESYS ориентирована на серийно выпускаемые изделия. Изготовитель ПЛК приобретает стартовый набор. Это комплекс из программного обеспечения и работ по обучению, помощи в адаптации и дальнейшему сопровождению. На выходе получается специальная «прошивка», «заточенная» под конкретную систему и готовая к тиражированию. Первая адаптация обычно занимает несколько месяцев. Выполнив ее, компания приобретает необходи-

мый опыт и может самостоятельно устанавливать CODESYS на любые свои продукты достаточно быстро, даже если они построены на разных процессорах и в разных операционных системах.

2. Существуют компании (Systec, Janz, Frenzel Berg и др.), предлагающие готовые встраиваемые устройства с CODESYS и системы под заказ. Заказчику остается только написать прикладное ПО. Обычно такие компании выпускают собственный ряд модулей «полуфабрикатов». У них имеется надежное аппаратное ядро (встраиваемый компьютер, микропроцессорный модуль, PLC Core), определенный набор плат или микросхем ввода/вывода, сетевые и другие модули. Из них komponуется нужная система. Они также предлагают несколько типов готовых встраиваемых компьютеров (контроллеров) с CODESYS и эволюционные наборы.
3. Применение микросхем и модулей Beck IPC@CHIP. Это миниатюрный встраиваемый компьютер с ОС PV на борту. Компании Beck удалось придумать технологию и создать специальный инструмент — Platform Builder (кстати, бесплатный). С его помощью в диалоговом режиме мы задаем требуемую конфигурацию системы исполнения CODESYS. Например, можно включить поддержку CANopen, веб-визуализации, описать входы/выходы, выбрать способ обслуживания энергонезависимой памяти, добавить собственные обработчики системных событий и т. п. Затем

▲ Самолетный тягач компании TREPEL. Вместо обычных рычагов управления он оснащен встраиваемым панельным PC с CODESYS компании Janz Tec



▲
Пульт управления
яхтой, разработанный
компанией Echo

▼
«Беспилотный»
транспортёр E&K
AUTOMATION на базе
собственного встроенного
контроллера и модулей
ввода/вывода Wago IO

автоматически генерируются все необходимые файлы. Остается дописать по готовым шаблонам драйверы ввода/вывода под нашу периферию и собрать систему исполнения. Получается исполняемый файл, который копируется на встроенный диск IPC@CHIP. Технология выглядит простой, но пока никто из конкурентов не создал аналогов. Все они предлагают некие типовые сборки PLC Core ядер с фиксированным функционалом.

По требованию российских заказчиков Beck создала специальное исполнение чипов с расширенным температурным диапазоном (-40°C). Существует исполнение для энергетики с поддержкой коммуникационной библиотеки МЭК 61850.

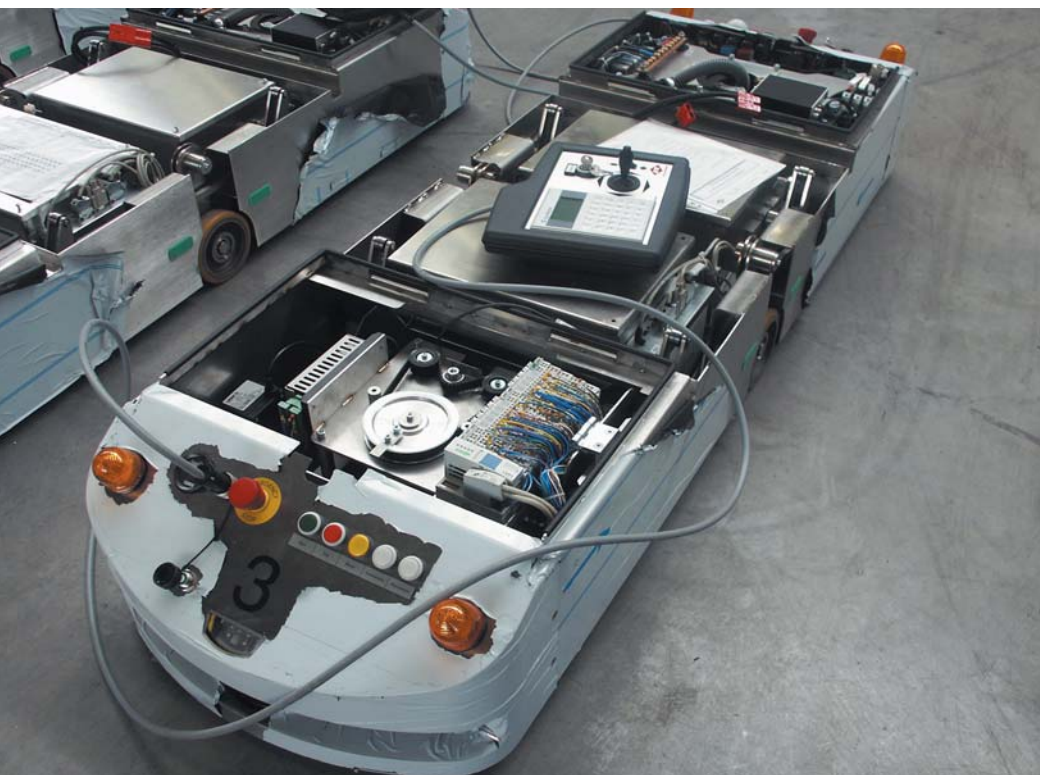
Первый путь выбирают крупные изготовители встраиваемых систем. Он оправдан при выпуске от нескольких сотен изделий в год и выше. В странах ЕС все более развивается практика заказа разработки. По числу применений в России лидирует технология Beck IPC@CHIP. В любом случае среда программирования CODESYS поставляется бесплатно. Никаких ограничений в функционале и числе установок в ней не предусмотрено. В CODESYS имеется встроенный эмулятор контроллера. Это позволяет начать работу без приобретения аппаратных средств.

ЗАКЛЮЧЕНИЕ

Сегмент встраиваемых систем в суммарном годовом объеме применений CODESYS ежегодно увеличивается. CODESYS применяется во встраиваемых контроллерах компаний Bosh, Rolls-Royce Marine, Praxis, CC Systems, Moba и др. Это

далеко не опытные прототипы, речь идет о десятках тысяч изделий. Примеры нескольких применений показаны на фотографиях.

Среди МЭК-систем программирования CODESYS выделяется тем, что, подобно компиляторам C/C++, непосредственно генерирует надежный и компактный машинный код, пригодный для встраиваемых систем. Простые в освоении языки МЭК позволяют привлечь к разработке и сопровождению специалистов прикладной области. Интерес для разработчика встраиваемых систем может представлять богатый функционал комплекса CODESYS. Многозадачность реального времени, обработка событий, встроенная визуализация, развитый набор коммуникаций, «горячее» обновление кода, полевые сети, поддержка управления через Интернет, средства национальной локализации проектов и другие функции CODESYS могут быть не востребованы во встраиваемой системе изначально. Но необходимо учитывать, что все они создавались эволюционно, исходя из практических требований, возникавших у пользователей контроллеров в разных странах, разных условиях и на разных этапах работ. В процессе жизни встраиваемой системы неизбежно возникают аналогичные или близкие задачи. Например, задача настройки и тестирования оборудования заказчиком, интеграция с другим оборудованием, веб-интерфейс и т. п. Во многих случаях CODESYS даст готовое решение. ●



ЛИТЕРАТУРА:

1. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования. М.: СОЛОН-Пресс. 2004.
2. Болл К. История возникновения программируемых логических контроллеров // Control Engineering Россия. 2009. № 1(36).
3. Петров И. В. Выбор ПЛК: ставка на открытость и совместимость // Конструктор. Машиностроитель. 2013. № 1.
4. Wagner R., Petrov I. Go to OOP // SPS-Magazin. 2012. № 9.
5. Петров И. В. Язык ST для С программиста // Мир компьютерной автоматизации. 2006. № 1.
6. Харазов В. Г. Интегрированные системы управления технологическими процессами. СПб.: Профессия. 2009.
7. Денисенко В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. М.: Горячая линия—Теленком. 2009.
8. Petry J. IEC61131-3 mit CoDeSys V3. Ein Praxisbuch für SPS-Programmierer. Kempten, Germany. 2011.
9. Константинов А. Модульный ПЛК FASTWEL I/O — от замысла до реализации // Control Engineering Россия. 2012. № 4(41).