

Virtual PLC – конец эпохи промышленного компьютера?

Юрий Широков

Виртуальный ПЛК меняет привычную архитектуру промышленной автоматизации, отделяя функцию управления от специализированного аппаратного контроллера. В статье рассматривается влияние виртуализации на детерминизм и работу с удалённым I/O, вопросы отказоустойчивости, функциональной и кибербезопасности. Уделено внимание консолидации вычислительных ресурсов, интеграции AI и цифровых двойников, а также жизненному циклу программного контроллера.

Вступление

Виртуальный ПЛК/PLC постепенно заставляет менять представление о привычной архитектуре промышленной автоматизации, отделяя функцию управления от специализированного аппаратного контроллера. На первый взгляд это может показаться очередным этапом развития Soft PLC: управляющая программа просто переносится с отдельного PLC на промышленный компьютер. Но на практике изменения гораздо глубже. В традиционной системе аппаратный PLC является одновременно вычислительной платформой, носителем ядра PLC runtime (среда выполнения, исполняющая программу пользователя, написанную на языках стандарта IEC 61131-3), коммуникационным узлом и средством взаимодействия с вводом-выводом (I/O). В архитектуре Virtual PLC эти функции могут быть разделены. Управляющая логика превращается в программный экземпляр, который выполняется на промышленном компьютере, Edge-устройстве, сервере или виртуальной инфраструктуре, а физические входы и выходы могут быть пространственно отдалены от вычислительного узла и подключены к нему по промышленной сети (рис. 1).

Siemens, например, предлагает SIMATIC S7-1500V как аппаратно-независимый Virtual PLC, разворачиваемый в Industrial Edge, а Beckhoff развивает TwinCAT Runtime for Linux и возможность запуска

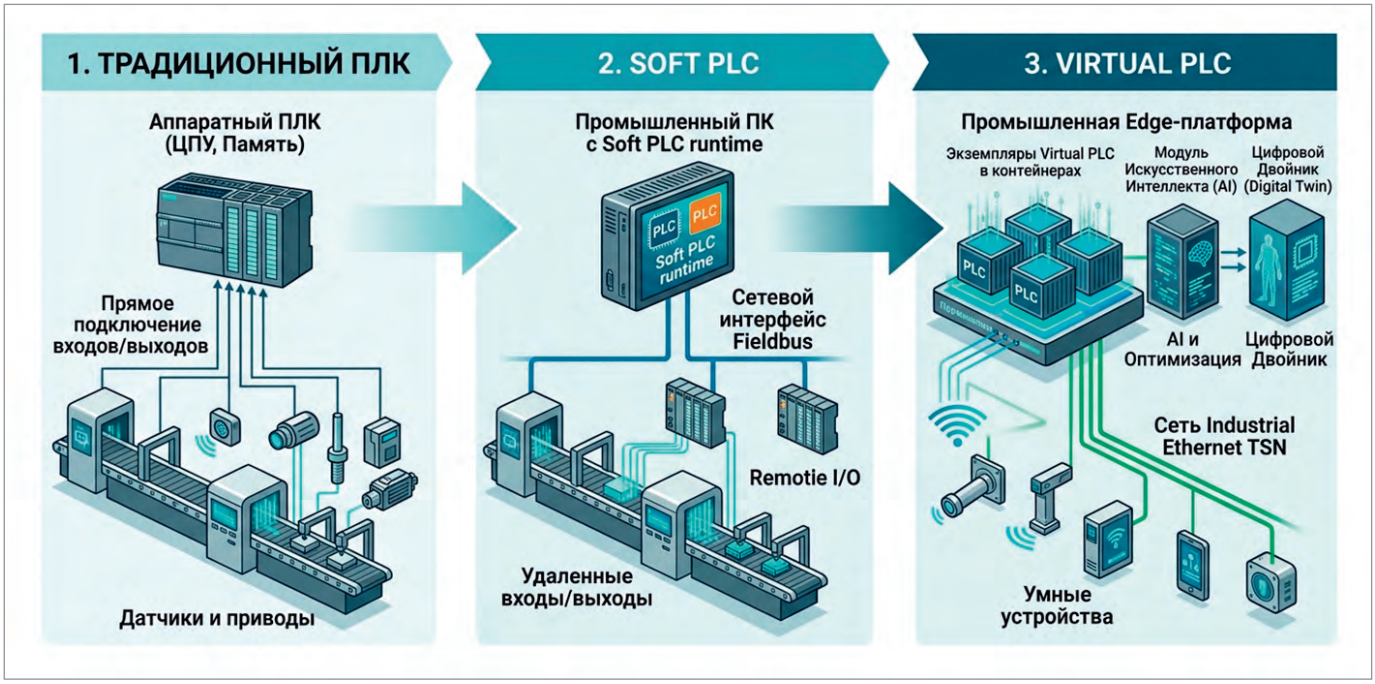


Рис. 1. Эволюция архитектуры управления: от аппаратного PLC к Virtual PLC и программно-определяемой автоматизации

нескольких runtime на одном Industrial PC, в том числе с использованием контейнеров. Международный стандарт IEC 61131-1 (отечественный аналог – ГОСТ Р МЭК 61131-1-2016) при этом давно допускает выполнение функций PLC не только на специализированной аппаратной платформе, но и на компьютере общего назначения с промышленными характеристиками. Поэтому главный вопрос будущего промышленной автоматизации состоит в характеристиках и свойствах вычислительно-коммуникационной платформы, на которой должна выполняться функция управления. Итак, мы начинаем переход от классического PLC к Software-Defined Automation.

Где граница

Исторически PLC создавался как специализированная вычислительная машина для управления технологическим процессом: в нём есть процессор, память, программная среда выполнения, коммуникационные интерфейсы, средства диагностики и механизмы взаимодействия с периферией. Все эти элементы объединены в одном физическом устройстве, и такой подход имеет очевидные преимущества.

Аппаратная платформа заранее известна и хорошо протестирована, производитель контролирует и программную среду, в результате инженер получает относительно законченный продукт с предсказуемыми характеристиками. Именно эта целостность и предсказуемость десятилетиями были одной из главных причин успеха PLC.

Но есть тут и несомненные минусы. Если требуется ещё один контроллер, приходится приобретать ещё один аппаратный контроллер. Если не хватает вычислительной производительности, выбирается другая модель PLC. Если нужна новая функция, приходится ориентироваться на возможности конкретного семейства. У такого подхода есть и ещё одна особенность: вычислительный ресурс каждого контроллера обычно оказывается привязан к конкретной функции управления. Таким образом, если на предприятии одновременно работают десять PLC, это не означает, что каждый из их процессоров постоянно загружен. Как следствие, значительная доля вычислительных ресурсов большую часть времени может оставаться невостребованной. IT-инфраструктура давно научилась решать проблему оптимизации ресурсов с помощью виртуализации и консолидации: один сервер может одновременно выполнять несколько приложений, виртуальных машин и контейнеров, распределяя вычислительные ресурсы между задачами. И вот концепция Virtual PLC переносит эту идею в промышленную автоматизацию.

Но здесь возникает принципиальное отличие от офисных приложений. Для обычного IT-приложения задержка выполнения в несколько миллисекунд или десятков миллисекунд часто не имеет практического значения, но в системе управления технологическим процессом сплошь и рядом важна не только средняя производительность, но и гарантированное время реакции, иными словами – детерминированность. Поэтому организацию Virtual PLC нельзя рассматривать как простой перенос программного кода с «железного» PLC на более мощный компьютер, так как виртуализируется вычислительная функция, но не физическая сущность технологического объекта, где датчик температуры остаётся датчиком, клапан – клапаном, привод по-прежнему должен получить команду с допустимой задержкой, сеть по-прежнему имеет задержку и джиттер, а технологический процесс может требовать перехода в безопасное состояние при отказе вычислительной платформы. Поэтому виртуализация не только меняет устройство PLC, но и смещает границы самой системы управления.

Soft PLC, Virtual PLC и контейнеры

Вокруг программных контроллеров уже возникла терминологическая путаница. Soft PLC, Virtual PLC, Software PLC и containerized PLC иногда используются почти как синонимы. Но правильнее рассматривать их как разные характеристики программно определяемого контроллера (рис. 2).

Soft PLC определяет, где реализована функция PLC. Это программная реализация контроллера, работающая на вычислительной платформе вместо специализированного PLC CPU. Такой контроллер может выполняться на промышленном компьютере под Windows, Linux или другой поддерживаемой операционной системой. При этом Soft PLC

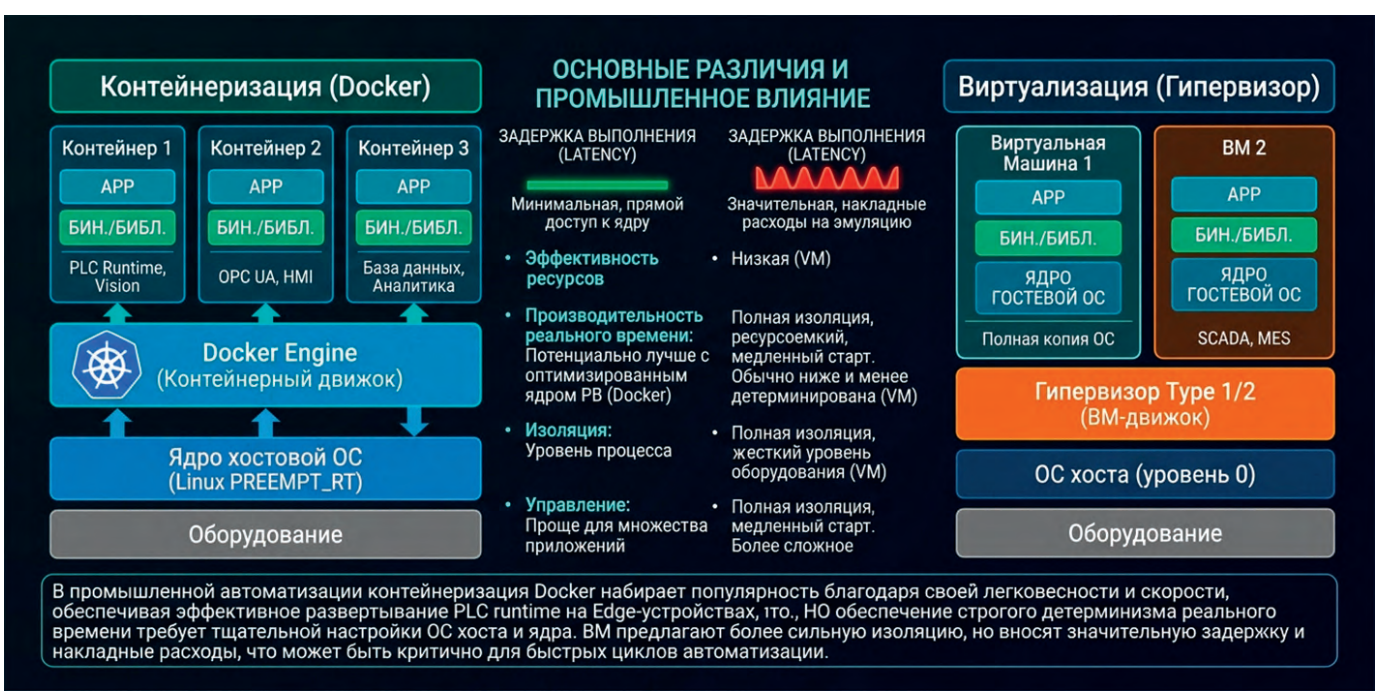


Рис. 2. Битва концепций: контейнеры vs виртуальные машины

ещё не обязательно означает полную независимость от «железа». Конкретная программная реализация может быть жёстко связана с определённым IPC (промышленным компьютером), операционной системой, драйверами или аппаратным интерфейсом.

Virtual PLC описывает более высокий архитектурный уровень абстракции: насколько функция управления отделена от конкретного физического контроллера и насколько свободно она может быть размещена в универсальной вычислительной инфраструктуре. К примеру, Siemens непосредственно использует термин Virtual PLC для S7-1500V, называя его аппаратно-независимым контроллером и предоставляя его как приложение Industrial Edge. Компания также подчёркивает совместимость с TIA Portal и централизованное управление экземплярами Virtual PLC.

Контейнеризация же отвечает совсем за другое: каким способом программный компонент упаковывается, изолируется и разворачивается. PLC runtime может находиться внутри контейнера вместе с необходимыми для работы и специально протестированными на совместимость библиотеками и конфигурациями. Это упрощает воспроизводимое развёртывание, обновление, управление версиями и перенос приложения между совместимыми средами. Поэтому контейнеризированный PLC не является отдельной от Virtual PLC сущностью.

На практике все эти подходы вполне могут сочетаться. Например: на Industrial PC под Linux в контейнере работает PLC runtime. Или: на Industrial Edge в виртуальной среде работает Virtual PLC + HMI + аналитика. Или: на Industrial PC размещено несколько контейнеров, а в них несколько PLC runtime + vision + AI.

Кстати, Beckhoff уже предлагает именно такую комбинацию: TwinCAT Runtime for Linux поддерживает контейнерные технологии Docker, Podman и LXC, а несколько TwinCAT Runtime могут выполняться на одном промышленном компьютере. Компания связывает это с аппаратной консолидацией и более эффективным использованием вычислительной мощности.

Контейнер и детерминированность

Термин «контейнеризированный PLC» легко создаёт ложное ощущение промышленной надёжности – такая самодостаточная программная коробочка, однако контейнер не является гарантией реального времени, поскольку он использует ядро хостовой операционной системы, а его временны́е характеристики зависят от всей системы: планировщика, конфигурации CPU, драйверов, механизмов изоляции ресурсов, сетевого стека, самого PLC runtime и аппаратной платформы. Но из этого и не следует, что контейнер и real-time несовместимы. Beckhoff, например, использует собственный дистрибутив Linux с real-time kernel для TwinCAT Runtime for Linux и заявляет стабильное real-time управление, одновременно поддерживая Docker, Podman и LXC. Иными словами, детерминизм обеспечивается не самим контейнером, а специальной комбинацией runtime, ядра, аппаратуры и настроек системы.

Таблица 1. Сравнение контейнеризации с VM

Критерий	Контейнер	Виртуальная машина
Основной уровень абстракции	Пользовательское пространство ОС	Виртуализированная аппаратная среда
Собственное ядро ОС	Как правило, нет	Да
Накладные расходы	Обычно ниже	Обычно выше
Скорость запуска	Обычно высокая	Обычно ниже
Изоляция	Зависит от конфигурации ядра и runtime	Обычно более выраженная на уровне V
Управление ресурсами	Средствами ОС и container runtime	Гипервизором и гостевой ОС
Реальное время	Не гарантируется самим контейнером	Не гарантируется самой VM
Типичное преимущество	Модульность и быстрое развёртывание	Изоляция и независимость программной среды

Виртуальная машина создаёт виртуализированную аппаратную среду, в которой может работать собственная гостевая операционная система. Контейнер же изолирует приложение и его пользовательское пространство, при этом используя ядро хостовой ОС (табл. 1).

Но для промышленной автоматизации вопрос выбора нельзя свести к простому сравнению в пользу контейнера или VM, так как возможна и комбинированная архитектура, когда контейнер разворачивается внутри виртуальной машины. Такой подход позволяет разделить различные классы приложений, но одновременно добавляет новые программные уровни. Следовательно, задача инженера заключается в поиске такой комбинации механизмов изоляции, которая обеспечивает требуемые предсказуемость, отказоустойчивость, безопасность и удобство эксплуатации.

Доменная архитектура

Вполне возможно, что наиболее интересная архитектура будущего промышленного компьютера будет выглядеть как модель нескольких изолированных вычислительных доменов (рис. 3).

Фактическая реализация будет зависеть от конкретной платформы: это могут быть отдельные runtime, контейнеры, виртуальные машины, привязка процессов к процессорным ядрам, специализированные ядра realtime и другие механизмы. Суть архитектуры заключается в другом. Разные типы вычислительных задач должны получать разные ресурсы и гарантии: PLC нужен предсказуемый runtime, AI нужна вычислительная мощность, Vision может требовать GPU, Historian нужен стабильный доступ к накопителю, OPC UA и другие коммуникационные сервисы требуют сетевых ресурсов, Cybersecurity требует изоляции и контроля доступа. И всё это может использовать одну физическую вычислительную платформу. Следовательно, промышленный компьютер перестаёт быть просто местом для запуска PLC. Он становится вычислительной системой управления.

И снова о времени

Предположим, PLC должен выполнять технологическую задачу каждые 10 мс. В традиционном контроллере производитель «держит в руках» практически всю вычислительную цепочку: PLC task → runtime → CPU → I/O. В Virtual PLC она может выглядеть значительно сложнее: PLC task → runtime → scheduler → → virtual CPU → ОС/гипервизор → physical CPU. А если I/O находятся удалённо, сюда добавляется: runtime → network stack →

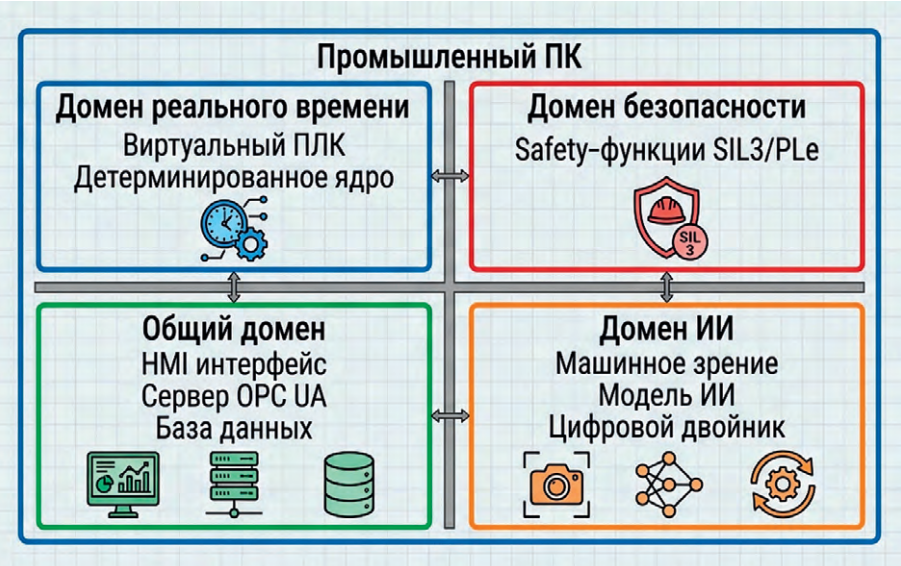


Рис. 3. Многодоменная архитектура вычислительной платформы

→ industrial network → remote I/O → physical process.

И теперь задержку формирует далеко не один контроллер. У системы появляются как минимум три разных характеристики, которые нельзя путать: производительность – сколько вычислений может выполнить система в единицу времени; задержка – сколько времени проходит от события до реакции; детерминизм – насколько предсказуемым является это время. Поэтому компьютер с очень мощным процессором может выполнять огромное количество вычислений в секунду и одновременно быть неподходящим для определённого контура управления, если максимальная задержка (worst-case latency) или джиттер (jitter) выходят за допустимые пределы.

В связи с этим для Virtual PLC необходимо оценивать не только производительность CPU и объём памяти, но и worst-case latency, jitter, политику планирования, распределение ядер CPU, доступ к памяти и I/O, сетевые задержки, поведение системы при перегрузке и способность к восстановлению после отказа.

И вот яркий пример. В собственных технических материалах Siemens для S7-1500V указывает real-time capability на базе механизмов стандартного Linux, поддержку нескольких экземпляров PLC на одном устройстве и PROFINET RT с циклом 1 мс. Одновременно документация Siemens перечисляет ряд отличий от аппаратного S7-1500, в том числе отсутствие I/O send clock ниже 1 мс и поддержку IRT (Isochronous Real-Time – режим обмена данными через интерфейс PROFINET) не в том же объёме, что у аппаратных CPU. Поэтому даже внутри одного семейства слова «реальное время» не должны рассматриваться как универсальная гарантия для любой задачи.

В классическом PLC промышленная сеть часто воспринимается прежде всего как отдельно стоя́щее средство обмена данными. В Virtual PLC эта концепция меняется. Если порты I/O физически отделены от вычислительной платформы, сеть становится частью временно́го контура. При этом сам переход на Ethernet не означает автоматического появления детерминизма. Даже специализированная промышленная сеть работает в составе конкретной инфраструктуры, где важны конечные устройства, коммутаторы, конфигурация трафика, синхронизация и механизмы восстановления. Интересно в этом отношении развитие OPC UA FX. OPC Foundation уже завершила этап Controller-to-Controller, а в 2026 году продолжает работу над Controller-to-Device (C2D), включая интеграцию remote I/O, приводов, датчиков и исполнительных устройств. В сентябре 2026 года на PlugFest OPC UA FX несколько производителей продемонстрировали взаимодействие контроллеров и прототипов I/O и приводов, в том числе с использованием TSN. При этом отдельные элементы C2D-архитектуры всё ещё проходят практическую отработку.

Промышленная сеть движется к более унифицированной коммуникационной архитектуре, но инженер всё равно должен анализировать временны́е характеристики всей системы, а не одного протокола.

А что с надёжностью?

Здесь, пожалуй, главное слабое место этой концепции. Если пять физических устройств заменить одним Industrial PC, мы действительно уменьшаем количество аппаратных компонентов, но одновременно усугубляем последствия отказа единой платформы. Если один сервер выполняет функции PLC № 1, PLC №2, HMI, vision, OPC UA и historian, то при отказе сервера мы потенциально лишаемся всех этих функций одновременно. В терминах архитектуры высокой доступности увеличивается так называемый blast radius (буквально «зона поражения»), или масштаб последствий одного отказа. Поэтому возникает логичный вопрос, ответ на который скорее всего никому не понравится и заставит задуматься о резервировании Industrial PC: какая часть технологического процесса будет парализована при отказе одного вычислительного узла? Но и резервирование ещё не гарантирует безопасности, так как необходимо понимать, можно ли восстановить состояние процесса после сбоя. Восстановить выполнение программного образа сравнительно легко, а вот физическую модель объекта в текущем моменте восстановить значительно сложнее. Если сервер перезапустился, он должен откуда-то получить актуальное состояние клапанов, двигателей, изделий на конвейере, накопителей, рецептур, последовательностей технологического цикла и всех остальных физических параметров, имеющих значение для процесса. Поэтому необходимо разделять три разные задачи: восстановление программного компонента, восстановление доступности системы управления и восстановление безопасного состояния технологического процесса.

Virtual PLC предоставляет новые механизмы резервирования и восстановления, но не отменяет инженерную разработку отказоустойчивой архитектуры.

Естественно было бы предположить, что критические функции безопасности будут продолжать выполняться только отдельным Safety PLC, а Virtual PLC останется исключительно инструментом стандартного управления.

Но современная практика показывает, что эта граница уже начала смещаться. Тот же Siemens предлагает SIMATIC S7-1500V F – fail-safe-вариант виртуального PLC. Компания сообщает о TÜV-сертификации решения и его использовании в производственной системе Audi. В актуальном каталоге Siemens для CPU 1517VF также указываются функциональные уровни безопасности SIL 3 по IEC 61508 и PL e по ISO 13849. Это

принципиально важный факт, показывающий, что виртуализация сама по себе не исключает выполнение safety-функций. Но из этого совершенно не следует, что любой Virtual PLC автоматически может стать Safety PLC: ведь функциональная безопасность является свойством конкретной сертифицированной архитектуры.

Здесь необходимо учитывать:

- аппаратную платформу;
- software stack;
- runtime;
- средства диагностики;
- сетевые соединения;
- safety I/O;
- механизмы обнаружения отказов;
- процедуры разработки;
- верификацию;
- валидацию.

Siemens подчёркивает, что при создании S7-1500V F рассматривался не только виртуальный CPU, но и вся система, включая взаимодействие с failsafe-периферией и механизмы обнаружения и компенсации отказов.

Это позволяет сформулировать важный принцип: виртуализация не отменяет safety engineering (инженерию безопасности), а переносит её на другой архитектурный уровень.

Кибербезопасность

Здесь важно не смешивать два направления. Functional Safety защищает человека, оборудование и технологический процесс от опасных отказов системы. Cybersecurity наоборот защищает систему от несанкционированного доступа и намеренного воздействия человека. В Virtual PLC они переплетены теснее, нежели в классическом PLC, у которого аппаратная граница между управляющей системой и IT-инфраструктурой часто достаточно очевидна. В Virtual PLC эта граница становится лишь программной. И если на одной вычислительной платформе одновременно находятся PLC runtime, HMI, AI, историческая база данных и сетевые сервисы, каждая из этих функций становится потенциальной частью общего фронта атаки. Защищать приходится уже не только PLC-программу, необходимо контролировать:

- операционную систему;
- hypervisor или container runtime;
- сетевые интерфейсы;
- учётные записи;
- сертификаты;
- образы приложений;
- механизмы обновления;
- резервные копии;
- каналы удалённого доступа.

При этом виртуализация может улучшить кибербезопасность, так как AI-приложение можно изолировать от PLC, разным функциям можно назначить разные права, ограничить сетевые связи, запретить аналитическому приложению прямой доступ к критическим каналам управления. Но успех не предопределён и будет зависеть уже не от самого факта виртуализации, а от продуманности архитектуры системы в целом.

Бок о бок с AI

Виртуализация становится особенно интересна с появлением промышленного искусственного интеллекта. Если PLC отделён от конкретного hardware и превращён в программную функцию, рядом с ним значительно проще разместить вычислительные компоненты AI. PLC и AI тогда выполняют разные функции: PLC реализует детерминированную логику управления и ограничения, AI может обнаруживать закономерности, прогнозировать состояние оборудования, классифицировать объекты, выявлять аномалии или рассчитывать оптимизационные рекомендации.

Например, AI-модель может определить повышение вероятности отказа узла и предложить изменение режима эксплуатации. PLC при этом может принять решение только после проверки рекомендации по заданному диапазону. Такой алгоритм значительно реалистичнее полной передачи управления нейронной сети. И проблема заключается даже не в том, что AI обязательно работает недетерминированно в техническом смысле, а в том, что поведение сложной модели гораздо труднее формально предсказать и ограничить относительно всех возможных недопустимых состояний технологического процесса.

Разумеется, в дальнейшем, по мере совершенствования моделей AI, граница этого разделения может и поменяться.

Виртуализация даёт ещё одно огромное преимущество, выходящее за рамки обычной эксплуатации. Если PLC является программным объектом, его значительно проще связать с цифровой моделью оборудования. Архитектура может выглядеть так:

Digital Twin → Virtual PLC → виртуальные I/O → модель технологического процесса.

Это позволяет перейти к Software-in-the-Loop (тестированию и валидации программного обеспечения в смоделированной виртуальной среде). PLC-программа выполняется в виртуальной среде и взаимодействует не с реальным оборудованием, а с его моделью. Следующий шаг – Hardware-in-the-Loop. Здесь реальная вычислительная платформа PLC взаимодействует с моделью технологического процесса, работающей в режиме реального времени. Получается последовательность:

Digital Twin → SIL → HIL → Virtual PLC → → физическое производство.

Такой подход может существенно упростить пусконаладку поскольку часть программных ошибок можно найти до подключения оборудования. Сценарии аварий в нём можно воспроизводить многократно, можно проверять взаимодействие нескольких машин, исследовать редкие комбинации событий, которые трудно специально создать на работающем объекте. Для сложного производства это становится не просто удобным инструментом разработки, а способом радикально уменьшить стоимость физических испытаний.

Жизненный цикл

У классического PLC жизненный цикл выглядит примерно так:

купить → установить → запрограммировать → запустить → обслуживать → → заменить.

Для Virtual PLC последовательность меняется:

разработать → смоделировать → протестировать → собрать программный образ → развернуть → настроить → → мониторить → обновлять → управлять версиями.

Это очень похоже на управление программным продуктом, тем не менее напрямую переносить DevOps из IT в промышленную автоматизацию нельзя. В корпоративном ПО, например, обновление приложения может быть выполнено ночью практически незаметно для пользователя. На непрерывном производстве такой фокус не пройдёт так как обновление PLC runtime потребует остановки непрерывного процесса, проверки технологического состояния и формальной процедуры допуска. Поэтому правильнее говорить не о DevOps «в лоб», а об адаптации принципов управления программным жизненным циклом к требованиям OT. При этом Virtual PLC действительно увеличивает значимость контроля версий и конфигураций. Под контролем должны находиться уже не только исходные PLC-программы. Необходимо управлять также версией ОС,

драйверами, сетевой конфигурацией, сертификатами, учётными данными, конфигурацией ресурсов, резервными копиями. Это одновременно преимущество и новая сложность: Virtual PLC делает контроллер легче переносимым, но увеличивает количество программных зависимостей, которые необходимо контролировать.

В традиционном PLC изменение firmware или программы относится к конкретному физическому устройству, но в Virtual PLC обновление может стать централизованной операцией. И это уже используется в Industrial Edge: Siemens подчёркивает возможность централизованного управления виртуальными контроллерами и программными обновлениями. На первый взгляд, это очевидное преимущество, но возникает обратная сторона: если одна ошибка обновления затронет сто виртуальных PLC, проблема распространится сразу на сто производственных участков. Поэтому масштабируемость Software-Defined Automation одновременно является её сильной и слабой стороной. Чем проще развернуть изменение, тем важнее контроль того, что именно будет развёрнуто. Другими словами, промышленная автоматизация начинает перенимать не только преимущества IT, но и его проблемы.

Ещё одна особенность программного PLC заключается в изменении самой модели владения. Имея дело с классическим контроллером, инженер в первую очередь думает о цене аппаратуры. В виртуальной архитектуре появляется существенная составляющая в виде набора лицензий на различные программные функциональные блоки, например лицензия PLC runtime, лицензия Industrial Edge, лицензия HMI, лицензия AI и т.п.

Программная модель позволяет масштабировать количество экземпляров в зависимости от потребностей, и Siemens, например, прямо подчёркивает возможность изменять количество используемых Virtual PLC и управлять ими централизованно. Но при расчёте TCO (Total Cost of Ownership – совокупная стоимость владения) необходимо учитывать не только аппаратную экономию, но и весь жизненный цикл лицензий. В промышленной автоматизации, где оборудование может эксплуатироваться десятилетиями, это становится особенно важным. Нужно заранее понимать, что произойдёт при окончании подписки, как работает резервирование, можно ли восстановить систему через десять лет, как осуществляется миграция на новую платформу. Для IT это привычные вопросы, но для промышленного предприятия с 20-летним жизненным циклом оборудования они приобретают принципиально иное значение.

От «железа» к Virtual PLC

Полный одномоментный отказ от традиционных контроллеров на всём предприятии является максимально неразумной стратегией. Гораздо реалистичнее поэтапная миграция. Первый шаг – не переносить наиболее критический контур. Для пилота лучше выбрать участок, где консолидация действительно может дать пользу, но отказ вычислительной платформы не приведёт к тяжёлым последствиям. Второй шаг – перенести на Industrial Edge или IPC второстепенные функции, не связанные напрямую с управлением.

Например HMI, историзацию, OPC UA, диагностику, аналитику. На этом этапе



предприятие начнёт получать опыт эксплуатации общей IT/OT-платформы без изменения основного контура управления. Третий шаг – установить Soft PLC или Virtual PLC рядом с действующей системой в качестве дублёра. На этом этапе можно сравнить различные параметры работы двух параллельных систем. Четвёртый шаг – перенести remote I/O и проверить реальную работу распределённого контура. Пятый – организовать резервирование вычислительной платформы. Шестой – провести SIL/HIL-тестирование новых программных конфигураций. И только после этого имеет смысл начинать консолидацию нескольких PLC на одном физическом узле.

В разговоре о software-defined automation легко забыть, что нижний уровень промышленной системы остаётся физическим. Это датчики, приводы, двигатели, роботы, I/O и т.д. Ни контейнер, ни виртуальная машина, ни AI не отменяют эти элементы. Поэтому внедрение Virtual PLC нельзя рассматривать как превращение АСУ ТП в обычный датацентр. Верхние уровни действительно становятся всё более программными, но нижний уровень продолжает подчиняться законам физического мира. Именно здесь проходит одна из главных границ архитектуры будущего.

Теперь вернёмся к исходному вопросу: закончилась ли эпоха промышленного компьютера? И ответ будет: скорее наоборот. Промышленный компьютер становится ещё важнее, поскольку его роль меняется. Раньше IPC часто рассматривался как специализированный компьютер для конкретной задачи. В новой архитектуре он превращается в универсальную вычислительную платформу, на которой одновременно могут находиться и Virtual PLC, и несколько PLC runtime, и HMI, и OPC UA, и AI. То есть Virtual PLC не вытесняет промышленный компьютер, а делает его центральным физическим элементом программно-определяемой автоматизации.

Но чем более виртуальным становится PLC, тем важнее становится физическая платформа, на которой он работает. Чем больше функций консолидируется на одном IPC, тем серьёзнее становятся требования к резервированию. Чем больше программных компонентов появляется, тем важнее управление версиями и кибербезопасность. Чем дальше PLC находится от I/O, тем важнее характеристики (в частности, детерминированность) сети. Чем ближе AI сливается с контуром управления, тем важнее изоляция его функций от детерминированной логики.

Таблица 2. Преимущества Virtual PLC в различных сценариях применения

Сценарий	Потенциальное преимущество	Что необходимо проверить
Машиностроение и сборочные линии	Тиражирование программных конфигураций	Цикл управления, motion и I/O
Роботизированные ячейки	PLC, vision и AI на одной платформе	GPU, latency и safety
Модульное производство	Быстрое создание одинаковых экземпляров	Изоляция и управление версиями
Конвейерные системы	Централизованное управление распределёнными участками	Задержки и отказ сети
Тестовые стенды	Быстрая реконфигурация и воспроизводимость	Реальные I/O и HIL
Цифровые двойники	Связь модели и управляющего ПО	Корректность модели
Большое количество однотипных машин	Консолидация и централизованное управление	Blast radius отказа
Системы с интенсивной аналитикой	AI, vision и PLC на одном IPC	Изоляция ресурсов
Простая автономная машина	Не всегда очевидны преимущества	TCO и сложность эксплуатации
Жёсткие safety-контуры	Возможна виртуализация сертифицированных функций	Соответствие конкретной safety-архитектуре

Получается своеобразная смена точки сложности, перемещающая её с уровня отдельного устройства на уровень архитектуры.

Ради чего такие сложности?

Почему промышленному предприятию вообще может понадобиться Virtual PLC? Уж точно не потому, что виртуальность сама по себе является преимуществом. Главный практический аргумент – возможность объединить вычислительные функции, которые раньше выполнялись на нескольких отдельных устройствах.

Предположим, на одном производственном участке имеются:

- PLC;
- HMI;
- OPC UA server;
- historian;
- машинное зрение;
- AI inference;
- сервис диагностики;
- цифровой двойник.

Традиционно это набор отдельных компьютеров и контроллеров. При Software-Defined Automation значительная часть этих функций может выполняться на одной производительной Industrial Edge-платформе.

Здесь и появляется экономический эффект, благодаря которому потенциально может уменьшиться количество физических устройств, упроститься тиражирование программных конфигураций, появится возможность эффективнее использовать CPU, облегчается централизованное управление программными версиями, упрощается масштабирование одинаковых машин и производственных модулей. Но оговорка «потенциально» здесь принципиально важна – Virtual PLC не гарантирует автоматическое приобретение системой всех названных преимуществ. А кроме этого, к вычислительной платформе добавляются требования к резервированию, сетевой инфраструктуре, кибербезопасности, лицензированию и квалификации персонала. Поэтому реальная экономическая модель должна рассматриваться через TCO, а не только через цену PLC CPU.

В традиционной системе значительная часть расходов приходится на аппаратную инфраструктуру:

- PLC;
- модули I/O;
- сетевое оборудование;
- IPC;
- шкафное оборудование;
- резервные устройства.

У Virtual PLC часть расходов перемещается из hardware в software и инфраструктуру. То есть появляются другие статьи:

- Industrial PC или сервер;
- гипервизор или Edge runtime;
- лицензии PLC;
- лицензии HMI;
- резервная вычислительная платформа;
- системы хранения и резервного копирования;
- средства кибербезопасности;
- обслуживание ОС и программного стека.

Поэтому в первую очередь надо определить, позволяет ли консолидация на конкретном производственном участке уменьшить совокупную стоимость владения при сохранении необходимых характеристик надёжности и управления. И здесь результат будет зависеть от задачи. Для простой автономной машины с одним контроллером традиционный PLC будет оставаться совершенно рациональным решением. Для производства с большим количеством однотипных машин и многочисленными вычислительными сервисами преимущества консолидации могут оказаться гораздо заметнее. Но особенно привлекательна модель там, где один вычислительный узел способен одновременно выполнять несколько реально востребованных функций. Универсального ответа для всех объектов здесь нет. В таблице 2 отражены потенциальные преимущества применения Virtual PLC в различных ситуациях. Особенно интересны системы, где вычислительных функций много, а физических машин большое количество. Но если объект представляет собой одну простую установку с несколькими I/O и одним PLC, Virtual PLC может создать больше архитектурной сложности, чем решить проблем.

В этом нет противоречия, ведь новая технология не обязана заменять старую везде. Она становится ценной там, где её архитектурные свойства максимально соответствуют задаче. Не надо рассматривать виртуализацию как универсальное лекарство от проблем современной автоматизации. Virtual PLC сам по себе не решает проблему детерминизма, он не устраняет задержки сети, не создаёт функциональную безопасность автоматически, не защищает систему от кибератак одним фактом того, что PLC работает в виртуальной среде, не устраняет единую точку отказа при консолидации, не заменяет плохую архитектуру I/O, не делает AI предсказуемым, не избавляет предприятие от необходимости иметь квалифицированный OT-персонал. И наконец, выбор Virtual PLC не означает, что весь промышленный процесс можно перенести в серверную или облако: нижний уровень АСУ ТП всё равно остаётся физическим, и это принципиальное ограничение всей концепции.

Заключение. Контроллер больше не коробка

В классической АСУ ТП слово «контроллер» обозначало вполне конкретный физический объект. Его можно было взять в руки, установить в шкаф и заменить. В программно-определяемой системе контроллер всё больше становится логической функцией вычислительной платформы. Преимущество этого подхода заключается не в том, что программное обеспечение принципиально лучше аппаратуры. Его преимущество – в гибкости: можно создать новый экземпляр устройства, изменить распределение вычислительных ресурсов, перенести функцию на другую совместимую платформу, централизованно управлять версиями, рядом с PLC разместить AI, vision, historian и цифровой двойник, создать виртуальную копию системы для тестирования. Но за эту гибкость приходится платить новыми механизмами изоляции, новыми процедурами резервирования, жёсткими требованиями к кибербезопасности, новыми подходами к лицензированию, наконец – новыми компетенциями на стыке OT и IT.

Эпоха PLC не заканчивается. Заканчивается, возможно, другая эпоха – эпоха, в которой функция промышленного управления обязательно была привязана к отдельному специализированному физическому контроллеру. PLC становится программной функцией, но это не означает, что промышленная автоматизация превращается в обычную IT-инфраструктуру. В промышленности по-прежнему существуют worst-case latency, безопасное состояние, функциональная безопасность, физические I/O, электромагнитные помехи, реальные двигатели и реальные люди. Именно поэтому Virtual PLC нельзя воспринимать как простую замену «железной» коробки коробкой программной – это изменение архитектурной философии.

Литература

1. IEC 61131-1:2003. Programmable controllers – Part 1: General information. IEC, 2003.
2. IEC 61131-3:2025. Programmable controllers – Part 3: Programming languages. IEC, 2025.
3. IEC 61508-1:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements. IEC, 2010.
4. IEC 62443-3-3:2013. Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels. IEC, 2013.
5. IEC 62443-4-2:2019. Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components. IEC, 2019.
6. How Real (Time) Are Virtual PLCs? // 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). DOI: 10.1109/ETFA52439.2022.9921545.
7. Characterizing the Real-Time Communication Performance of Virtual PLC in Industrial Edge Platform. IEEE, 2025.
8. Siemens. SIMATIC S7-1500V Virtual PLC – Software PLC.
9. Siemens. S7-1500 Virtual Controller CPU 1517V(F). Function Manual. V2.0.1, 01/2025.
10. Beckhoff. TwinCAT Runtime for Linux. Technical documentation and product materials.
11. Linux Kernel Documentation. Real-time preemption (PREEMPT_RT).
12. OPC Foundation. OPC Unified Architecture – Part 80: UAFX Overview and Concepts.
13. OPC Foundation. OPC Unified Architecture – Part 81: UAFX Connecting Devices and Information Model.
14. A Digital Twin Framework for Programmable Logic Controllers in Industrial Automation Systems. Procedia CIRP, vol. 139, 2026, pp. 446–452. DOI: 10.1016/j.procir.2025.08.193.

НОВОСТИ реклама НОВОСТИ реклама

Новая серия панельных компьютеров AdvantiX PPC-EA

Компания AdvantiX представляет новую серию панельных компьютеров PPC-EA, которая является логическим развитием серии PPC-ES. Все модели PPC-EA поставляются с резистивными сенсорными экранами диагональю 15–21", опционально доступны к заказу модели и с ёмкостными сенсорными экранами. А также можно заказать устройства с экраном повышенной яркости и контрастности. Модели серии защищены от попадания воды и пыли по стандарту IP66 по передней панели и могут эксплуатироваться при температурах от +5°C до +40°C. Данная серия построена на высокопроизводительном современном процессоре Intel Core i3-1215U (6 ядер, 1,2 ГГц/4,4 ГГц в турбо-режиме) с возможностью опционально установить более производительный Intel Core i5-1235U (10 ядер, 1,3 ГГц/4,4 ГГц в турбо-режиме).



Все модели PPC-EA уже по умолчанию комплектуются производительной оперативной памятью уровня DDR5. На задней панели доступен широкий набор портов ввода/вывода: 2 порта USB 3.2 type A, 2 порта USB 2.0 type A, 2 сетевых порта RJ45, два последовательных порта RS-232/422/485 и один RS-232. Также, уже в базовой версии, идут три COM-порта. ●