

Развиваем архитектуру АСУ ТП:

от стандарта IEC 61499 к готовому решению на платформе 4DIAC и Ethernet APL



В статье рассмотрены технические и структурные решения, разрабатываемые компанией «ИНСОЛ»: среда разработки 4DIAC, построенная в соответствии со стандартом IEC 61499, а также вычислительный и сетевой хаб INSOL NODE и интеллектуальный распределенный контроллер INSOL 1000, образующие единую платформу для построения систем автоматизации без шкафов автоматики.

ООО «ИНСОЛ», г. Уфа

Архитектура, построенная в строгом соответствии со стандартом IEC 61499, фундаментально меняет традиционный подход к построению АСУ ТП (рис. 1). Ее ключевая особенность – **полное отсутствие централизованных шкафов автоматики**. Вместо них интеллектуальные устройства распределяются по объекту, что сокращает

затраты на кабельную инфраструктуру и монтаж. В операторной требуется разместить лишь систему бесперебойного питания (ИБП) и коммутаторы для агрегации трафика.

Данная структура является экосистемой для распределенной автоматизации нового поколения. В разнообразии решений промышленной

автоматизации все острее ощущается потребность в комплексных решениях, которые объединяют вычислительную мощность, сетевое взаимодействие и гибкое управление в едином бесшовном пространстве. Ответом на этот вызов становится связка двух устройств – INSOL NODE и INSOL 1000, формирующая готовую

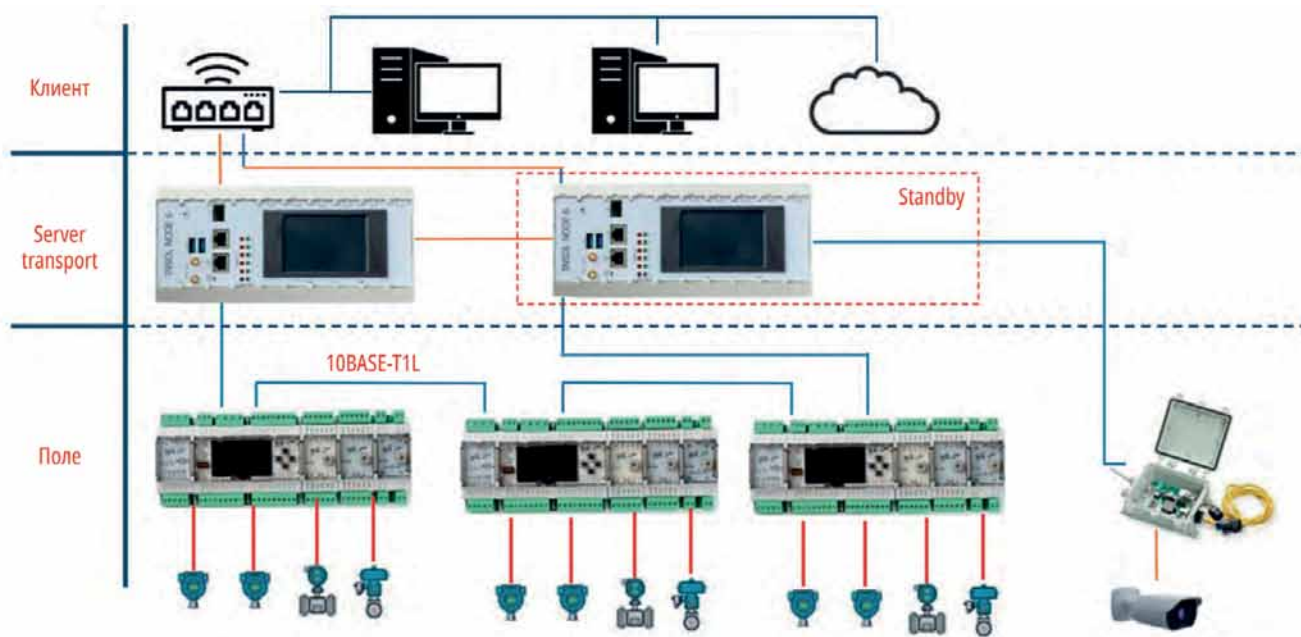


Рис. 1. Структура системы



Рис. 2. INSOL NODE и INSOL 1000

экосистему для построения сквозных систем IIoT и Industry 4.0 (рис. 2).

INSOL NODE: универсальный вычислительный и сетевой хаб

Что получится, если объединить в одном корпусе полноценный компьютер, управляемый коммутатор 2-го уровня, внешний накопитель на сотни гигабайт, базу данных с OPC UA-сервером на тысячу переменных, среду исполнения для распределенных ПЛК по стандарту IEC 61499 и встроенную веб-SCADA? Ответ — INSOL NODE. Это устройство выполняет роль мощного концентратора и шлюза верхнего уровня. Его ключевые компоненты раскрывают целый спектр возможностей.

► **Вычисления и хранение данных:** наличие полноценной вычислительной платформы позволяет запускать сложную логику, базы данных исторических трендов и аналитики, что выходит за рамки возможностей типичного программируемого контроллера.

► **Сетевое ядро:** встроенный управляемый коммутатор организует структурированную сетевую топологию, обеспечивая сегментацию трафика, приоритизацию данных и надежное соединение между всеми элементами системы.

► **Полевая сеть:** десяток изолированных портов Ethernet-APL с поддержкой питания (PoDL) по каждому каналу — это прямой выход в поле. Технология APL позволяет по двухпроводной линии передавать данные и питание на расстояние до 1500 м, что критически важно для подключения датчиков и исполнительных механизмов в условиях процессной автоматизации и взрывоопасных зон.

► **Единая программная среда:** интеграция OPC UA-сервера, веб-SCADA и среды выполнения IEC 61499 создает целостную среду для

разработки, управления и визуализации.

Инженер может в одном веб-интерфейсе настраивать проект как в режиме разработки (develop), так и в режиме исполнения (runtime) без необходимости установки дополнительного ПО.

INSOL 1000: интеллектуальный распределенный контроллер

Вторым базовым элементом системы выступает распределенный контроллер INSOL 1000. Его архитектура строится на общей шине Insol-BUS, к которой подключаются необходимые модули расширения (дискретные, аналоговые и другие). Адресация модулей задается механическим переключателем на лицевой панели, а система их автоматически распознает, максимально упрощая процесс конфигурации и замены.

Ключевой особенностью контроллера является присвоение физическим каналам именных алиасов. Вместо работы с малопонятными «сырыми» адресами (вроде `addr:0,channel:1,type:dout5`) инженер оперирует понятными именами, например: «Температура_Реактора_A». Эти алиасы напрямую доступны в среде разработки 4DIAC (реализация стандарта IEC 61499), что значительно ускоряет программирование и снижает количество ошибок.

INSOL 1000 также обладает развитыми коммуникационными возможностями.

► **Modbus-мастер:** два порта RS-485 работают в режиме мастера для опроса периферийных устройств (датчиков, частотных преобразователей). Переменные Modbus становятся равнозначными физическими каналами в среде 4DIAC, а конфигурация интерфейса производится через удобный веб-интерфейс.

► **Modbus TCP-клиент:** устройство может выступать в роли подчиненного Modbus TCP-устройства. В этом режиме внешние системы (например, главная SCADA завода) получают доступ как к данным физических входов контроллера, так и к переменным, созданным и вычисленным в логике 4DIAC.

Единая экосистема: синергия компонентов

Вместе INSOL NODE и INSOL 1000 образуют мощную и гибкую платформу. На текущий момент серия поддерживает ключевые промышленные протоколы:

- внутренние протоколы 4DIAC для распределенного управления;
- проприетарный протокол Insol-NET;
- Modbus TCP для интеграции со сторонними системами.

Находится в стадии разработки поддержка таких высокоскоростных протоколов, как Profinet и EtherCAT, что еще больше расширит область применения системы.

4DIAC: инструмент для бесшовной разработки распределенных систем управления

Среда разработки 4DIAC, являющаяся эталонной реализацией стандарта IEC 61499, предлагает интуитивно понятный и дружелюбный интерфейс для проектирования распределенных систем автоматизации. Основная работа ведется в **System Explorer** (рис. 3), который предоставляет доступ к трем ключевым элементам проекта:

- **Application** — создание основной логики программы;
- **System Configuration** — конфигурация аппаратного обеспечения;
- **Module Library** — хранилище программных блоков (как стандартных, так и пользовательских).

Процесс настройки оборудования интуитивно понятен и не требует написания сложных конфигурационных файлов. Инженер просто добавляет на рабочее поле Ethernet-устройства, такие как INSOL NODE и контроллеры серии 1000, после чего задает для каждого из них сетевые параметры: IP-адрес и порт (для устройств 1000-й серии обычно используется порт 61499). Важно отметить, что на INSOL NODE можно разместить несколько независимых приложений, работающих на

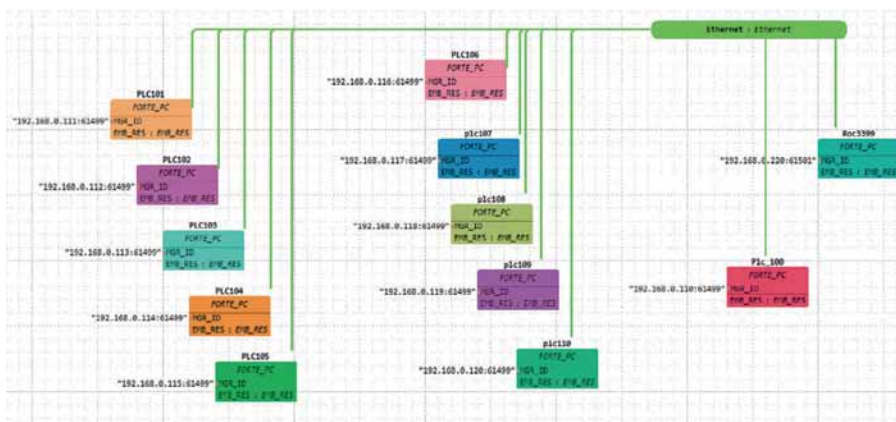


Рис. 3. System Configuration: визуальное проектирование аппаратной части

разных портах. Можно распределять нагрузку между приложениями следующим образом: одно выполняет реальные задачи, другое используется для отладки новых компонентов. Результатом этой настройки становится схематичное изображение всей системы, наглядно отображающее состав оборудования и сетевые связи между компонентами.

Application: визуальное программирование с событийной моделью

Непосредственное создание приложения управления заключается в компоновке программы из готовых блоков, доступных в библиотеке элементов. Основой для их работы служит событийная модель (Event-Driven) (рис. 4).

Принцип работы событийной модели:

- ▶ включается по событию – функциональный блок выполняет свою алгоритмическую функцию только при получении сигнала на свой вход события (EI или REQ);
- ▶ сообщает о завершении – по окончании вычислений блок генерирует выходное событие (EO или CNF), которое, в свою очередь, может запустить следующий блок в цепочке.

Структура функционального блока:

- ▶ входы/выходы событий определяют, когда блок выполняется и когда он завершает свою работу;
- ▶ входы/выходы данных отвечают за передачу значений переменных и состояний между блоками.

Итог: среда 4DIAC позволяет не только визуально сконфигурировать систему, но и создать распределенную программу, где логика четко разделена и выполняется строго по факту наступления определенных условий, что в полной мере соответствует принципам построения современных АСУ ТП.

Архитектура «Издатель – подписчик»: простота, скорость и визуальное программирование

В сетевом разделе библиотеки модулей представлены два типа функциональных блоков, заслуживающих особого внимания: издатель (PUBLISHER) и подписчик (SUBSCRIBE).

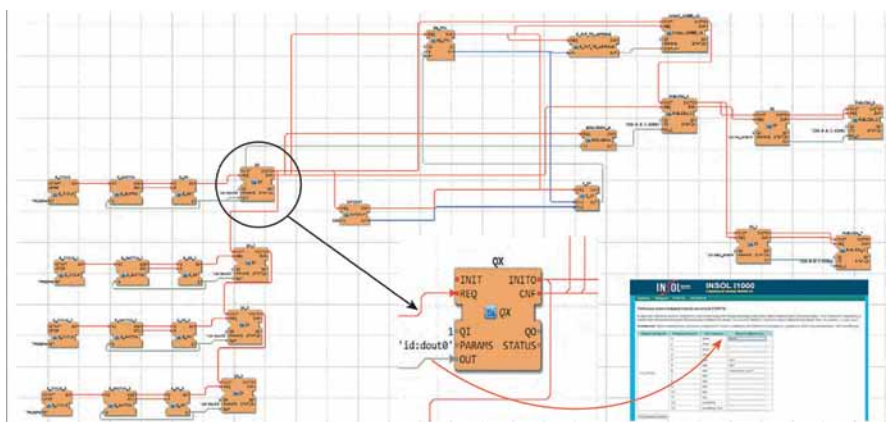


Рис. 4. Визуальное программирование с событийной моделью

Логика их работы проста и элегантна: блок-издатель передает данные по сети, используя внутренний идентификатор (ID), а неограниченное количество подписчиков может одновременно получать эти данные. Это формирует основу для построения масштабируемых и гибких решений.

При этом простота не достигается в ущерб производительности. Проведенный стресс-тест, в котором 30 устройств серии 1000 были соединены в последовательную цепочку (каждое устройство выступало и подписчиком, и издателем), показал впечатляющий результат: сквозная задержка распространения данных по всей цепочке не превысила 50 мс. Это доказывает, что данная архитектура идеально подходит для систем управления реального времени.

Разработчику следует помнить, что у высокой скорости передачи есть обратная сторона медали. 4DIAC использует UDP-соединение (без подтверждения доставки), и для критически важных событийных сигналов надо разрабатывать триггерную схему передачи данных.

Привязка к «железу» (Map to): после создания программной логики ее необходимо сопоставить с физическими устройствами. Это делается одним щелчком мыши с помощью команды Map to в контекстном меню. Элементы приложения автоматически окрашиваются в цвета тех устройств, за которыми они закреплены.

Инициализация и запуск (Deploy): важно подключить программу к инициализирующему событию. После этого выполняется команда Deploy – она загружает программу в ОЗУ целевых устройств и немедленно запускает ее на выполнение.

Онлайн-отладка: система предоставляет удобные инструменты для отладки в реальном времени. Инженер может вручную инициализировать события и в режиме онлайн отслеживать прохождение сигналов между блоками, что делает процесс поиска и исправления ошибок быстрым и наглядным.

Создание автономного образа (Boot File): после полной отладки программы создается Boot-файл. По сути, это XML-описание всего приложения: все экземпляры блоков, их параметры и связи. Этот файл записывается во флеш-память устройства и автома-

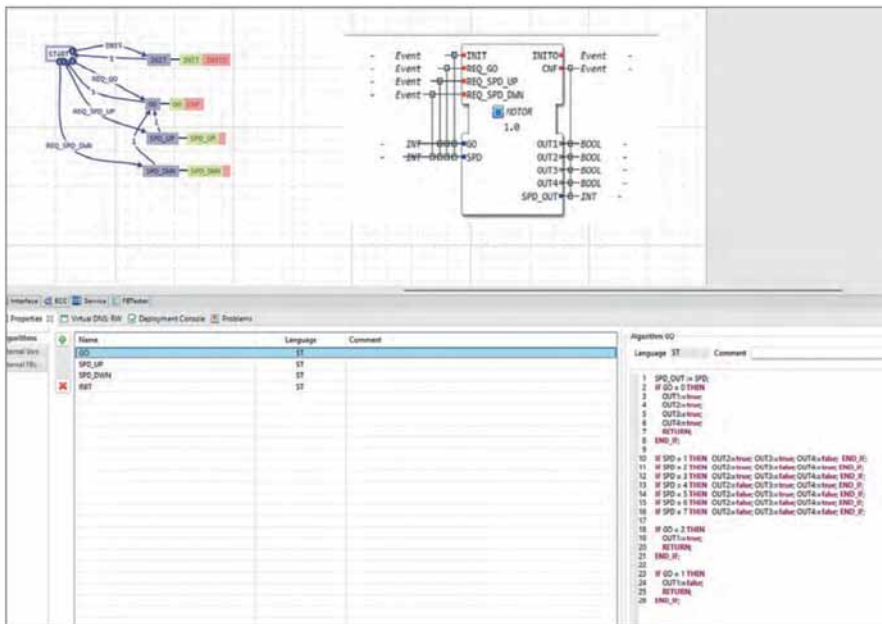


Рис. 5. Создание собственного функционального блока с 4 алгоритмами и текст одного из алгоритмов на ST

тически запускается при каждом его включении, обеспечивая полную автономность работы системы.

Создание собственных функциональных блоков

Любая мощная система должна позволять расширять свою функциональность. 4DIAC обладает комфортным и продвинутым инструментарием для создания пользовательских функциональных блоков (ФБ) (рис. 5).

Процесс их создания выглядит следующим образом. В интегрированной среде разработки (IDE) создается новый типовой ФБ, где описываются все его входные и выходные события и данные.

Типовой ФБ может содержать несколько алгоритмов, разработанных на привычном языке структурного текста (ST) стандарта IEC 61131-3.

Далее графически создается ЕСС-диаграмма (State Machine), которая

определяет, какой алгоритм и в какой последовательности будет выполняться в зависимости от того или иного события.

Система автоматически интерпретирует созданный пользовательский блок в код на языке C++.

Полученный файл C++ добавляется через веб-интерфейс INSOL NODE, где происходит его фоновая компиляция, полностью скрытая от пользователя.

После этого функциональность нового блока становится доступна во всех приложениях INSOL NODE.

На данном этапе развития продукта разработчики убрали возможность создания пользовательских блоков для модуля INSOL 1000, чтобы повысить отказоустойчивость, учитывая тот факт, что эти модули находятся «в поле» и оперативный доступ к ним может быть затруднен. Но создание простых ПИД-регуляторов, алгорит-

мов защиты и прочего доступно пользователю.

Интеграция в базу данных и OPC UA-сервер

Компания «ИНСОЛ» выпускает специальные функциональные блоки для записи и чтения данных различных типов во встроенную базу данных. Максимальное количество переменных для записи в БД ограничено 5000, при этом гарантированная скорость чтения/записи одной переменной не превышает 50 мс. Интеграция с внешними системами (такими как MES или ERP) осуществляется через OPC UA-сервер, который конфигурируется как отдельное приложение-надстройка над базой данных.

Визуализация: встроенная SCADA-система FUXA

Вишенкой на торте всей экосистемы является встроенная SCADA-система на базе FUXA. Это полностью веб-ориентированное решение, работающее как в режиме разработки, так и в режиме runtime. Система поддерживает графическую динамику, построение трендов, имеет собственную базу данных и может подключаться к данным через Modbus TCP или в качестве OPC UA-клиента к серверу, работающему на INSOL NODE.

Данный инструмент позволяет быстро, «здесь и сейчас», создавать и вводить в эксплуатацию мнемосхемы и панели оператора, что дает возможность использовать его для задач визуализации в составе комплексной платформы.

М. Р. Рафальсон, директор,
ООО «ИНСОЛ», г. Уфа,
тел.: +7 (347) 246-6024,
e-mail: info@insolsoft.com,
сайт: insolsoft.ru



Сейчас в СМИ

Все дублируется в новостной ленте Дзена

