

Рылов М.А., д.т.н. проф. Софиев А.Э.
МГТУ «МАМИ»
Rulmike@mail.ru

Аннотация. В статье рассматриваются вопросы интеграции информационной системы производства PI System и графического языка программирования LabVIEW. Рассмотрены механизмы взаимодействия систем. Приведены примеры программного кода.

Ключевые слова: PI System, PI SDK, Labview, интеграция.

PI (Plant Information) System – информационная система производства – разработанная компанией OSIsoft (США). PI System образует единую информационную платформу предприятия. Она позволяет охватить все технологические процессы, осуществляя сбор данных с любых PLC, DCS, SCADA, LIMS и их гарантированную доставку в единое хранилище (более 300 интерфейсов сбора данных). При этом осуществляется сжатие данных и их хранение в едином формате. Необходимо так же отметить что PI System имеет более 15 000 внедрений по всему миру и установлено на заводах крупнейших Российских компаний (ТНК-ВР, Лукойл, Газпром, Роснефть, Фосагро, Башнефть, Норильский Никель и других).

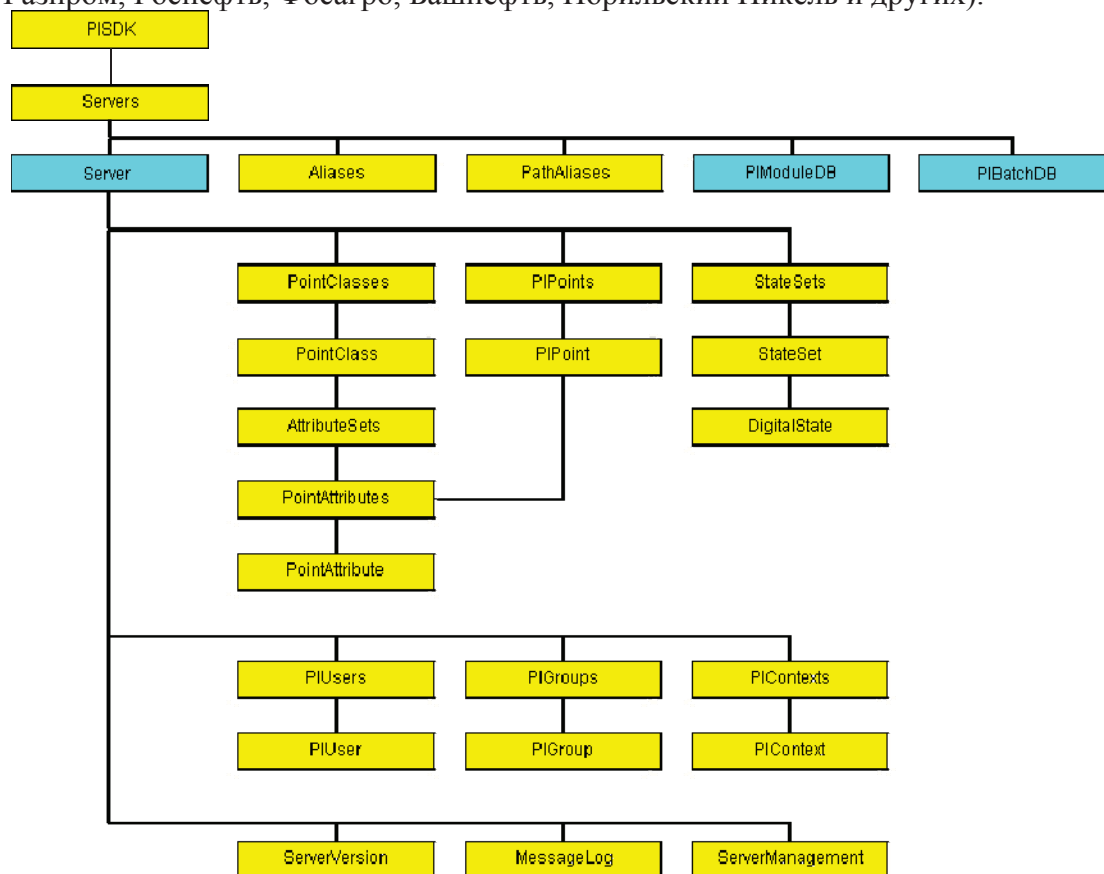


Рисунок 1. Объектная модель PI-SDK









“LabVIEW – это быстро развивающаяся среда графического программирования, которую используют технические специалисты, инженеры, преподаватели и ученые по всему миру для быстрого создания комплексных приложений в задачах измерения, тестирования, управления, автоматизации научного эксперимента и образования. В основе LabVIEW лежит концепция графического программирования – последовательное соединение функциональных блоков на блок-диаграмме” [4]. LabVIEW имеет огромное количество различных библиотек для моделирования, обработки сигналов и построения систем управления. Данная

среда позволяет с легкостью разрабатывать программное обеспечение с продвинутым графическим интерфейсом даже инженерам, которые не сильны в программирование на стандартных текстовых языках программирования.

Потому объединяя возможности PI System и LabVIEW можно разработать программное обеспечение для обработки и моделирования данных реальных промышленных процессов с наименьшими затратами на интеграцию и последующее внедрение. Переложив на PI System проблемы сбора и хранения данных, а также обеспечения безопасности при доступе к ним, на LabVIEW можно создать сложную математическую обработку и моделирование процессов.

Таблица 1

Основные функции LabVIEW для работы с .Net

Графическое обозначение	Название	Описание
	Constructor Node	Узел конструктора. Узел формирует экземпляр выбранного пользователем объекта .Net и выдает ссылку на него.
	Invoke Node	Узел метода. Вызывает метод класса переданного по ссылке.
	Property Node	Узел свойства. Читает или записывает свойство по ссылке, переданной на его вход.
	To More Generic Class	К более общему классу. Формирует ссылку к более общему классу в иерархии наследования.
	To More Specific Class	К более определенному классу. Формирует ссылку к более определенному классу в иерархии наследования.
	.NET Object to Variant	Преобразование типа данных из .Net в тип Variant (один из типов данных Labview, в который (из которого) могут преобразовываться любые другие типы Labview).
	To .NET Object	Преобразование типа данных из Variant в тип .Net.
	Close Reference	Закрывает ссылку. Закрывает ссылку на экземпляр объекта.

PI Software Development Kit

PI-SDK (PI Software Development Kit) – пакет средств разработки программного обеспечения PI. Представляет собой инструмент создания программ доступа к PI-серверам. PI-SDK основан на Microsoft's Component Object Model (Компонентная объектная модель – COM). PI-SDK обеспечивает объектно-ориентированный подход к программному взаимо-

действию с PI-системами. Предоставляет пользователю иерархическую модель объектов и их коллекций, которые соответствуют компонентам PI-серверов. Библиотеки доступны для разработки приложений для 64 битных и для 32 битных систем. А также позволяет вызывать библиотеки из языков класса Microsoft.Net. PI-SDK предоставляет доступ к следующим основным конфигурационным таблицам: PIPoints (БД Тегов), StateSets (Таблица дискретных состояний), PIUsers (Таблица пользователей), PIGroups (Таблица групп пользователей), PIModuleDB (Модульная БД) и другим.

Технология .Net также поддерживается средой LabVIEW. В таблице 1 перечислены основные функции для работы с .Net библиотеками.

Таким образом, используя технологию .Net в LabVIEW можно соединяться с PI-серверами, читать и записывать данные в теги.

Получение данных из тега PI-сервера в Labview.

На рисунке 2 показана блок-диаграмма основного прибора для считывания данных. Для того чтобы получить данные из тега, необходимо выполнить две основные операции:

Открыть соединение с PI-сервером; Считать данные из нужного тега.

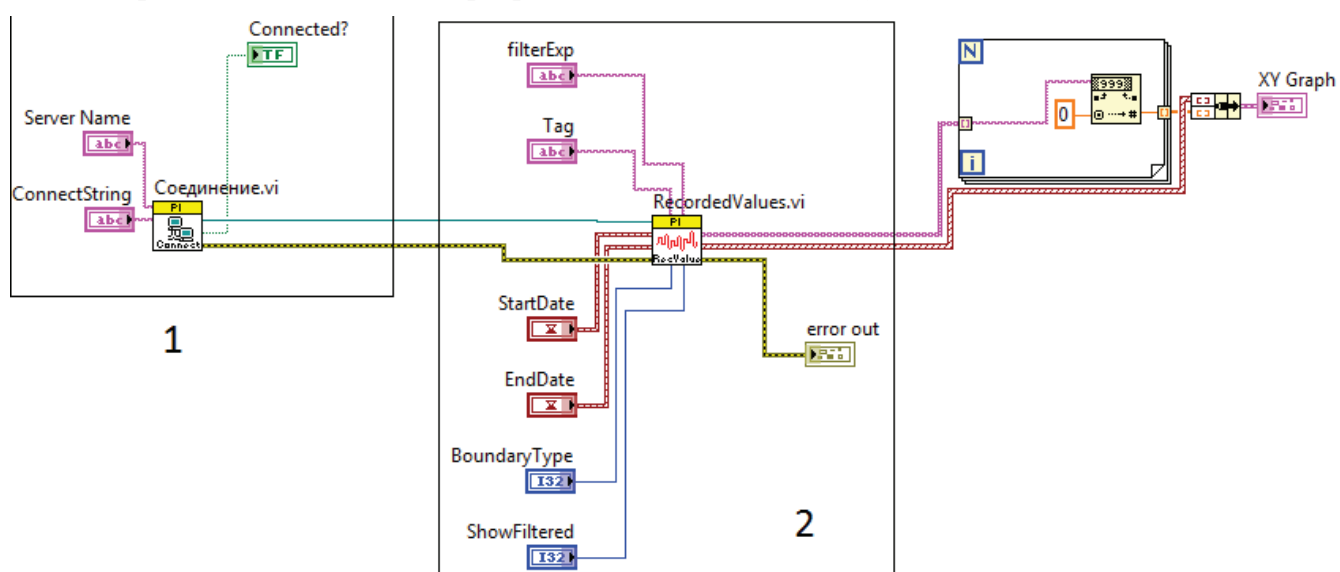


Рисунок 2. Блок-диаграмма основного прибора

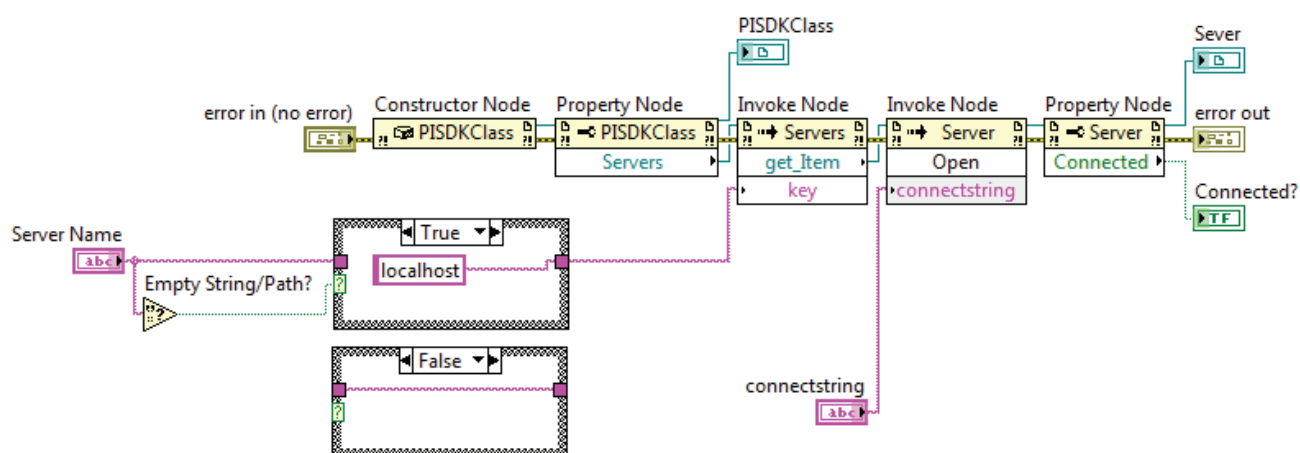


Рисунок 3. Блок-диаграмма ВП “Соединение.vi”

Рассмотрим каждую из этих операций более подробно.

Открытие соединения с PI-сервером

Для того чтобы открыть соединение с сервером PI надо вызвать метод Open объекта

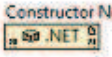

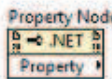





Server коллекции Servers (см. рисунок 1). На рисунке 3 представлена блок диаграмма ВП “Соединение.vi”, в таблице 2 описание элементов управления и индикаторов на блок диаграмме. Алгоритм работы ВП следующий: по имени сервера получается ссылка на него, затем вызывается метод Open и указывается строка для подключения, после чего с помощью свойства Connected проверяется успешность соединения.

Чтение данных тега

Существует довольно много методов чтения данных из сервера PI, но мы рассмотрим только один – RecordedValues. Данный метод возвращает архивные значения за указанный в запросе промежуток времени. Он присущ объекту PIData, который в свою очередь ассоциируется с объектом PIPoint. На рисунке 4 представлена блок диаграмма ВП “RecordedValues.vi”. А в таблице 3 описание элементов управления и индикаторов на блок диаграмме.

Таблица 2

Элементы блок диаграммы ВП “Соединение.vi”

Графическое обозначение	Название	Описание
	Constructor Node	Узел конструктора. Узел формирует экземпляр выбранного пользователем объекта .Net и выдает ссылку на него.
	Invoke Node	Узел метода. Вызывает метод класса переданного по ссылке.
	Property Node	Узел свойства. Читает или записывает свойство по ссылке, переданной на его вход.
	To More Generic Class	К более общему классу. Формирует ссылку к более общему классу в иерархии наследования.
	To More Specific Class	К более определенному классу. Формирует ссылку к более определенному классу в иерархии наследования.
	.NET Object to Variant	Преобразование типа данных из .Net в тип Variant (один из типов данных Labview, в который (из которого) могут преобразовываться любые другие типы Labview).
	To .NET Object	Преобразование типа данных из Variant в тип .Net.
	Close Reference	Закрывает ссылку. Закрывает ссылку на экземпляр объекта.

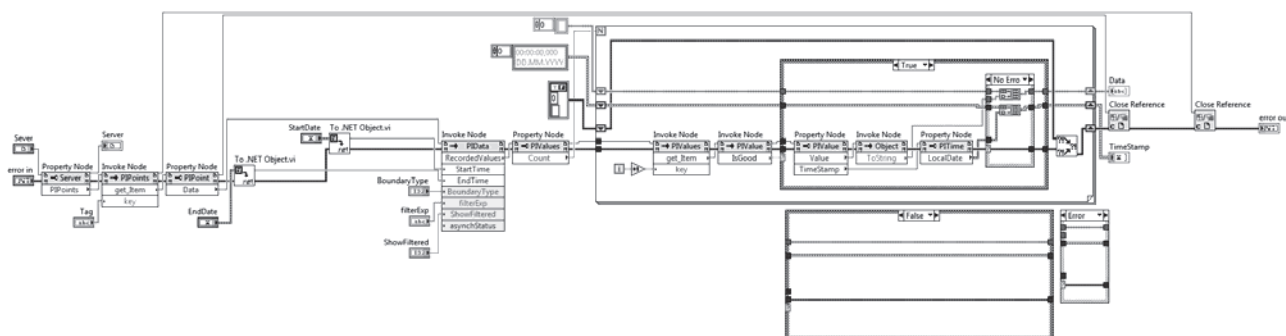


Рисунок 4. Блок диаграмма ВП “RecordedValues.vi”

Таблица 3

Элементы блок диаграммы ВП “RecordedValues.vi”

Графическое обозначение	Подпись	Описание
	StartDate	Начало промежутка времени за который нужны данные
	EndDate	Конец промежутка времени за который нужны данные
	BoundaryType	Определяет, как времена и значения возвращенных конечных точек определены. Возможны следующие варианты: 1) Inside (первое и последнее значение лежат внутри диапазона времени) 2) Outside (первое и последнее значения являются ближайшими точками лежащими вне диапазона времени) 3) Interp (крайние точки диапазона получаются путем интерполирования значений). 4) Auto (сервер сам выбирает какой вариант 1-3 использовать).
	ShowFiltered	Параметр, который управляет, возвращать ли фильтруемые значения или не показывать вообще. Является необязательным параметром.
	filterExp	Строка, содержащая выражение для фильтрации значений. Является необязательным параметром
	Tag	Названия тега
	error in	Ошибка с предыдущего ВП
	Sever	Ссылка на объект Sever из ВП “Соединение.vi”
	Data	Считанные значения тега (в строковом формате)
	TimeStamp	Временные метки значений тега
	error out	Ошибка на выходе ВП
	Sever	Ссылка на объект Sever (для использования в последующих ВП)

Алгоритм работы ВП следующий:

1. По имени тега получается ссылка на него;
2. Временные метки начала и конца интервала преобразуются в .Net формат;

3. Считываются данные тега;
4. В цикле происходит проверка считанных значений на правильность значений. Так как PI-сервер может помимо значений точек хранить и их статус (например, если не удалось получить значение или оно не правильного формата);
5. Если значение проходит проверку из пункта 4, то оно добавляется в массив результирующих значений. В противном случае пропускается;
6. Закрываются все ненужные ссылки.

Заключение

Таким образом, объединение двух систем дает ряд преимуществ:

- отпадает необходимость интеграции с разнородными системами завода, так как все данные хранятся в едином хранилище PI-сервера;
- широкое распространение PI System в нефтеперерабатывающей промышленности позволяет с легкостью переносить решения между заводами;
- использование встроенных механизмов безопасности данных позволяет отвечать современным требованиям;
- использование LabVIEW значительно снижает время разработки вычислительных алгоритмов, за счет использования стандартных математических библиотек.

Литература

1. Суранов А.Я. LabVIEW 8.20: справочник по функциям. -М.: ДМК Пресс, 2007.- 536 с.
2. П. Блюм LabVIEW: Стиль программирования. Москва.: ДМК Пресс, 2009. 400 с.
3. Рылов М.А. Программный комплекс для мониторинга и расчета показателей качества продукции в темпе с технологическим процессом. Сборник трудов 10 Международной научно-практической конференции «Инженерные, научные и образовательные приложения на базе технологий National Instruments - 2011», Москва 8-9 декабря 2011г. - М.: ДМК-пресс, 2011.
4. Официальный сайт компании National Instruments в России <http://www.labview.ru/>
5. Официальный сайт компании ООО «Индасофт» // <http://www.indusoft.ru>
6. Официальный сайт компании ООО «OSIsoft» // <http://www.osisoft.com>

Модель качества стабильного катализатора на установке каталитического риформинга бензина

Рылов М.А., д.т.н. проф. Софиев А.Э.
МГТУ «МАМИ»
Rulmike@mail.ru

Аннотация. В статье рассматриваются вопросы построения модели качества стабильного катализатора на установке каталитического риформинга бензина. Рассмотрен подход к сбору и предварительной обработке технологической информации. Приведены результаты моделирования для октанового числа стабильного катализатора. Сделано сравнение результатов моделирования и лабораторных данных.

Ключевые слова: каталитический риформинг, модель качества, стабильный катализатор, октановое число, факторный анализ, метод главных компонент.

Установка и процесс каталитического риформинга является обязательным звеном любого современного нефтеперерабатывающего завода. Назначение данного процесса – получение высокооктанового компонента автомобильных бензинов, ароматизированного концентрата для производства индивидуальных ароматических углеводородов и технического водорода в результате каталитических превращений бензиновых фракций.