ISaGRAF 6.1: ДИНАМИЧНОЕ РАЗВИТИЕ И КОНЦЕПТУАЛЬНЫЕ НОВШЕСТВА

А.В. Колтунцев, С.В. Золотарев (Компания ФИОРД)

Рассматриваются инновационные возможности современных контроллеров на базе новой версии системы программирования контроллеров ISaGRAF 6.1 с поддержкой международных стандартов IEC 61131-3 и IEC 61499. Новые возможности в ISaGRAF 6.1 ориентированы как на дальнейшее развитие базовой технологии программирования контроллеров, так и на поддержку нового направления – создания решений, удовлетворяющих требованиям функционально безопасных систем. Особое внимание в статье обращается на введение в ISaGRAF 6.1 новой концептуальной возможности – поддержки асинхронных событий и прерываний.

Ключевые слова: единая платформа автоматизации, плагины, интерфейс, прерывания, резервирование и восстановление, программирование контроллеров, функциональная безопасность.

Комплекс средств ISaGRAF [1–4] компании ISaGRAF Inc. широко известен как инструмент разработки приложений для ПЛК на языках стандарта IEC 61131-3 и IEC 61499, который позволяет создавать локальные или распределенные системы управления ТП. Основа технологии – среда разработки приложений ISaGRAF Workbench и адаптируемая под различные аппаратно-программные платформы исполнительная система ISaGRAF Runtime (Target, таргет, целевая задача). В ISaGRAF поддерживаются все пять языков стандарта IEC 61131-3: IL (Instruction List – Список инструкций), ST (Structured Text – Структурированный текст), LD (Ladder Diagram – Ступенчатая диаграмма), FBD (Function Block Diagram – Диаграмма функциональных блоков), SFC (Sequential Function Chart – Последовательная функциональная диаграмма) плюс языки FC (Flow Chart, Потоковая диаграмма, Блок-схема) и ANSI C.

Начиная с версии 6.0, ISaGRAF становится одной из компонент (CAM, «конкретных моделей») *Единой платформы автоматизации (ACP, Automation Collaborative Platform)*. Концепция и технология ACP разработана на основе ISaGRAF и создана для обслуживания систем автоматизации. Единая Платформа Автоматизации разработана как среда, управляемая с помощью открытых подключаемых модулей – плагинов и конкретных моделей автоматизации (CAM, Concrete Automation Model) (рис. 1). Плагины используются для конфигурирования и программирования, CAM – для интерфейса с таргетами (через интерфейс AAM –абстрактной модели автоматизации) [1]. Напомним, что разработка плагинов и CAM ведется с использованием Microsoft Visual Studio. В то время как разработка таргетов может вестись с помощью различных компиляторов, ориентированных на конкретные аппаратные средства и операционные системы. Однако ACP не является средой с открытым кодом. ACP представляет собой расширяемый слой абстракции с общим интерфейсом, который обеспечивает унифицированные функциональные возможности, выбираемые пользователем. ACP предназначена для поставщиков средств автоматизации, OEM-производителей, системных интеграторов, научно-исследовательских институтов. ACP помогает проектировщикам ПО, позволяя им сосредоточиться на своей основной предметной области,

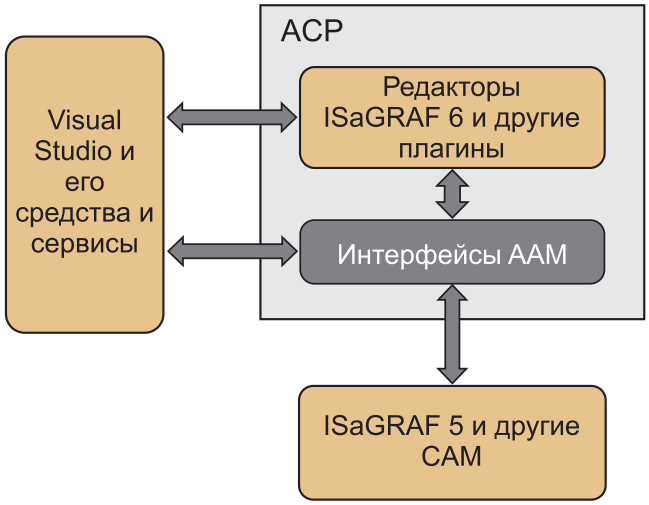

Рис. 1 Обобщенная архитектура ACP

а не на системных программных вопросах инфраструктуры решения. ACP поддерживает несколько моделей CAM одновременно, предоставляя возможность интеграции разнородных продуктов в единую интегрированную среду разработки. Две из конкретных моделей автоматизации, входящих в базовую поставку ACP, обеспечивают создание приложений для таргетов ISaGRAF 5 (CAM 5) и ISaGRAF 3 (CAM 3). Приложения в ISaGRAF 6 представляют собой виртуальные машины, работающие на различных аппаратных платформах, называемых исполнительными узлами (target nodes). Процесс разработки заключается в создания проекта, состоящего из устройств (devices), представляющих собой таргеты с одним или несколькими экземплярами ресурсов. Проекты могут разрабатываться, используя различные языки программирования, включая языки стандарта IEC 61131–3. После этапа разработки, ресурсы компилируются в TIC-код (target independent code) или в программу на языке C.

Одним из первых реализованных собственных плагинов в рамках технологии ACP стал плагин ISaVIEW, обеспечивающий пользователя простыми, но в то же время мощными интегрированными средствами человеко-машинного интерфейса (HMI). Страницы ISaVIEW встраиваются в структуру проекта автоматизации. Еще одним важным средством в рамках ISAGRAF 6 является инструментарий поддержки

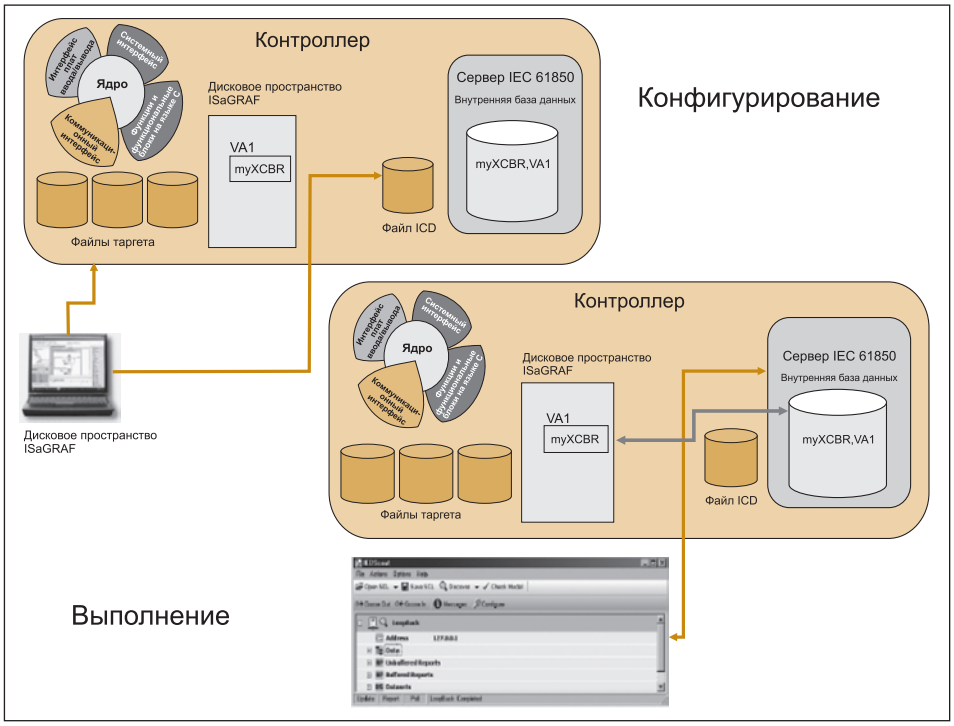

Рис. 2. Схема поддержки IEC 61850 в ISaGRAF 6

стандарта IEC 61850 «Сети и системы связи на подстанциях». В рамках ISaGRAF 6 поддерживаются все типы данных IEC 61850, но пользователь на этапе разработки своего приложения может с помощью утилиты Target Definition Builder for IEC 61850 выбрать, какие типы данных будут поддерживаться в его конкретной целевой задаче. Эти выбранные типы данных IEC 61850 пользователь сможет затем использовать в своих проектах на языках программирования, входящих в ISaGRAF 6, а на ISaGRAF-контроллере с поддержкой IEC 61850 будет доступна новая опция активизации генерации файла ICD (IED Capability Description), описывающего возможности микропроцессорных устройств. Кроме того, пользователь сможет в функциональных блоках IEC 61131 и IEC 61499 использовать переменные и входы/выходы типов данных IEC 61850. Архитектура решения, поддерживающего протокол IEC 61850, показана на рис. 2. На этапе загрузки Workbench передает сгенерированные файлы (TIC-код и ICD файл) соответственно целевой задаче и серверу IEC 61850. На этапе выполнения сервер IEC 61850 обеспечивает доступ к переменным ISaGRAF по запросам от клиентских задач.

В ISaGRAF 6.1 Workbench реализован редактор IEC 61499, а также поддерживаются языки LD, FBD, ST, SFC и SAMA (Scientific Apparatus Makers-Manufactures Association).

Новые возможности ISaGRAF 6.1

• Базируется на CAM 5.3[1]. Новая версия CAM 5.3 имеет несколько новых шаблонов, в том числе шаблоны Microsoft Windows и Linux для новых функций пользовательского прерывания и восстановления после отказа. К числу других новых характеристик относятся приоритеты SFC для одновременного перехода в случае OR-ветвления. В штатную поставку ISaGRAF 6.1 входят новые шаблоны для конечных пользователей, причем поддерживаются таргеты версии CAM 5.2, CAM 5.3 и CAM 3. Для CAM 5.3 наиболее продвинутыми являются следующие шаблоны: с поддержкой режима Failover (Windows, Linux), с поддержкой прерываний (Windows, Linux). В ISaGRAF 6.1 по-прежнему входит целевая задача ISaGRAF Free Target, работающая на платформе XP Embedded, 32-и 64-битной версии Windows XP, Windows Vista & Windows 7, и полностью функционирующая версия ISaGRAF 6.1 Workbench, с помощью которой разработчики могут создавать законченные приложения. ISaGRAF Free Target, включает драйверы Modbus TCP Client, а также Modbus TCP Server.

• Visual Studio 2010 и.NET 4.0. Среда ISaGRAF 6.1 Workbench базируется на мощном инструментарии Microsoft Visual Studio 2010 (вместо VS 2008, как это было в ISaGRAF 6.0), благодаря чему возросла скорость выполнения приложений. Кроме того, это помогло усовершенствовать пользовательский интерфейс и расширить средства управления пакетами.

• Механизм обработки прерываний.

• Дерево зависимостей. Плагин, реализующий дерево зависимостей (Dependency Tree), дает пользователям полный обзор всех связанных элементов в приложении, так что они могут видеть все зависимости переменных, а также восходящие и нисходящие зависимости для каждой переменной.

• Режим резервирования и восстановления (Failover).

• Библиотека блоков – это усовершенствование пользовательского интерфейса, которое дает пользователям возможность перетаскивать (drag and drop) любую функцию или функциональный блок в программу вместо осуществления выбора из селектора блоков. Функции и функциональные блоки являются контекстно-зависимыми от выбранного проекта, устройства или ресурса, они группируются по диапа-

[1] CAM 5.3 - модель, обеспечивающая создание приложений для таргетов ISaGRAF вер. 5.3.

зону или категории, и пользователи могут также выполнять поиск названия блока.

• Монитор блокировок переменных (Variable Lock Monitor) выводит перечень всех заблокированных переменных.

• Изменение IP адреса без компиляции проекта. Перекомпиляция потребуется только, если используется связывание переменных (binding) или режим Failover.

• Контроль версий исходных кодов (Version Source Control) обеспечивает поддержку совместной работы нескольких пользователей над одними и теми же элементами (например, устройство, ресурс, программный модуль – POU). Это также дает возможность пользователям работать с несколькими версиями проекта, создавать резервные копии и восстанавливать целые проекты или отдельные элементы проекта и сравнивать файлы, выполненные в разных версиях.

• Функция статуса контроллера (Controller Status) обеспечивает пользователей основной информацией о контроллере, включая версию проекта, с которой они работают, а также время цикла и число заблокированных переменных.

Режим резервирования и восстановления (Failover)

В ISaGRAF 6.1 реализована поддержка режима резервирования и восстановления после отказа. Восстановление после отказа – это режим работы, при котором функции системы управления принимает на себя вторичная система управления в том случае, когда главная система становится недоступной из-за отказа оборудования или при запланированном простое. Использование этой функции повышает отказоустойчивость системы управления. Функция восстановления после отказа в ISaGRAF 6.1 дает возможность пользователям модифицировать систему управления и менять условия, при которых контроллер получает или теряет управление. В режиме Failover среда ISaGRAF 6.1 реализует следующие ключевые возможности: безударный переход на резервную машину, работа с любым типом POU (SFC, FBD, LD, ST, 61499), автоматическая загрузка проекта (одновременно на первичную и вторичную станцию), автоматическое переключение из Workbench на нужную станцию в режиме отладки, автоматическое переключение OPC-серверов. В данный момент не реализована поддержка хранимых (retain) переменных.

Рассмотрим реализацию режима Failover в ядре исполнительной системы ISaGRAF. После загрузки приложения на активный контроллер автоматически выполняется его передача на резервный контроллер. Оба контроллера начинают параллельно выполнять одно и то же приложение. Входные переменные передаются из основного контроллера в резервный контроллер перед каждым циклом выполнения. В конце каждого цикла выполняется проверка по вычисленной контрольной сумме, чтобы гарантировать целостность данных и результатов. В случае расхождения на резервный контроллер передается вся область данных активного (основного) контроллера. При сбое на активном контроллере резервный контроллер становится активным (основным) и начинает управлять процессом. Для связи между основным и резервным контролером по умолчанию используется сеть TCP/IP, но могут быть использованы и другие сети (последовательный канал, UDP, оптоволокно). Допускается настройка условия, при котором происходит смена основного контроллера. Конфигурирование Failover состоит из установки следующих параметров: IP-адрес, номер порта и значения тайм-аутов.

Результаты тестирования режима Failover. Условия тестирования: 85 тыс. переменных пользователя, 4 тыс. входных переменных, объем TIC-кода равен 247k, размер буферов связи 64k, основной контроллер работает под управлением ОС Windows 7, резервный – под управлением VMware ОС Windows XP. Результаты тестирования: время цикла 1 мс с отключенным режимом Failover, время цикла 13 мс с включенным режимом Failover и полной синхронизацией данных (88102 байта в 2 фреймах), время цикла 6 мс с включенным режимом Failover и частичной синхронизацией данных (4127 байт в одном фрейме).

Прерывания – новое принципиальное новшество в ISaGRAF 6.1

В области автоматизации прерывание является асинхронным сигналом, указывающим на необходимость реакции на событие в аппаратном или программном обеспечении. Прерывание широко используется в приложениях, где необходимо малое время отклика и выполнения действия. Для каждого типа приложения необходим свой тип прерываний (время, импульс, высокоскоростные счетчики, ввод/вывод). Для поддержки различных приложений ISaGRAF предоставляет набор средств, позволяющий пользователю определить его собственные прерывания и сопоставить их с приложением ISaGRAF. Прерывание пользователя может поддерживать аппаратное прерывание (OEM-производителю необходимо реализовать конкретные системные функции для чтения приложением пользователя сигнала прерывания), прерывание от таймера, «программное прерывание» по адресу переменной (еще не реализовано).

Средства работы с прерываниями будут обеспечивать следующие возможности: поддержку до 256 прерываний (конфигурируемых), прерывания по таймауту по умолчанию, адаптацию на одно- и многозадачные исполнительные системы ISaGRAF, асинхронное выполнение (отдельные потоки или задачи), разрешение доступа к глобальным переменным через мьютексы[2] (семафоры). Средства поддержки прерываний будут входить в состав PRDK

[2] Мьютекс - вид семафора для работы с потоками.

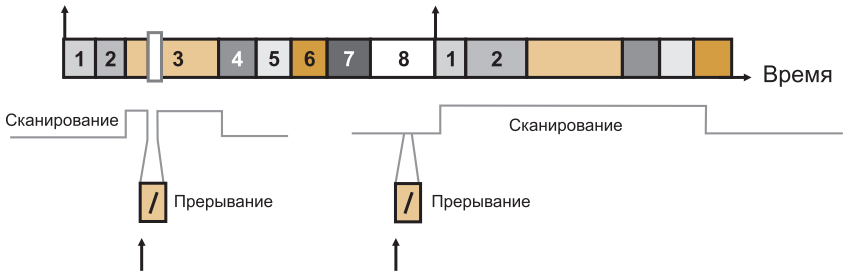

Рис.3

Tool Kit, а также будет поставляться плагин для конфигурирования прерывания и приложения пользователя. Ожидаемое время отклика на прерывание в рамках ISaGRAF 6.1 не должно превышать 100 мс.

Несколько слов о реализации концепции прерываний в целевой задаче ISaGRAF. Ядро системы исполнения ISaGRAF может портироваться как на однозадачную среду, так и многозадачную в зависимости от реализации OEM-производителем. При портировании в однозадачную среду ядро реализуется как поток (Thread). Эта архитектура подходит для всех моделей портирования.

Рассмотрим изменения в стандартном цикле ISaGRAF, которые появляются в связи с поддержкой прерываний. Обычный цикл выполнения приложения в ISaGRAF состоит из следующих этапов:

1) сканирование входов устройств;

2) прием (consume) значений связанных (bound) переменных;

3) выполнение TIC-кода приложения;

4) передача (produce) связанных переменных;

5) обновление (выдача) выходов устройств;

6) сохранение хранимых (Retained) значений;

7) обработка IXL-сообщений;

8) замирание (Sleep) до следующего цикла.

В ISaGRAF 6.1 приняты следующие соглашения (рис. 3):

• прерывание пользователя может быть вызвано на любом этапе цикла выполнения (обработки ввода/вывода, выполнения кода, обработки связывания и т. д.);

• прерывание пользователя не может вызываться в середине изменения TIC-кода, обрабатываемого по команде online-изменения. Прерывание будет запрещено перед вставкой/удалением кода и разрешено после завершения операции;

• прерывание пользователя может быть вызвано при работе целевой задачи в пошаговом режиме отладки. Когда режим выполнения возвращается в режим реального времени, таймеры будут сбрасываться.

Проектные решения, принятые для реализации прерывания в целевой задаче ISaGRAF:

1) приоритет ядра ISaGRAF должен быть меньше, чем прерывание пользователя;

2) при возникновении аппаратного прерывания последнее должно обращаться к мьютексу связанной с ним задачи для запуска алгоритма, определенного пользователем (рис. 4). Задача обработки прерывания пользователя должна конфигурироваться заранее для определенного POU (после разрешения семафором). Внутри задачи прерывания пользователя выполняется только код POU, то есть должно быть запрещено связывание или обработка ввода/вывода.

Прерывания определяются на стадии проектирования целевой задачи с помощью утилиты Target Definition Builder (TDB, Построитель определения таргета). Перед использованием в Workbench должны быть определены все прерывания. Полученный с помощью TDB файл с описанием целевой задачи с прерыванием, как правило, затем оформляется в виде шаблона, который уже может использовать конечный пользователь в рамках Workbench.

В Workbench прерывания определяются на уровне устройства, а конфигурируется на уровне ресурса. Для каждого нового устройства добавляется новый узел «прерывание» в Solution Explorer («Обозревателе решений») для каждого ресурса и новое окно «прерывания» (Interrupts).

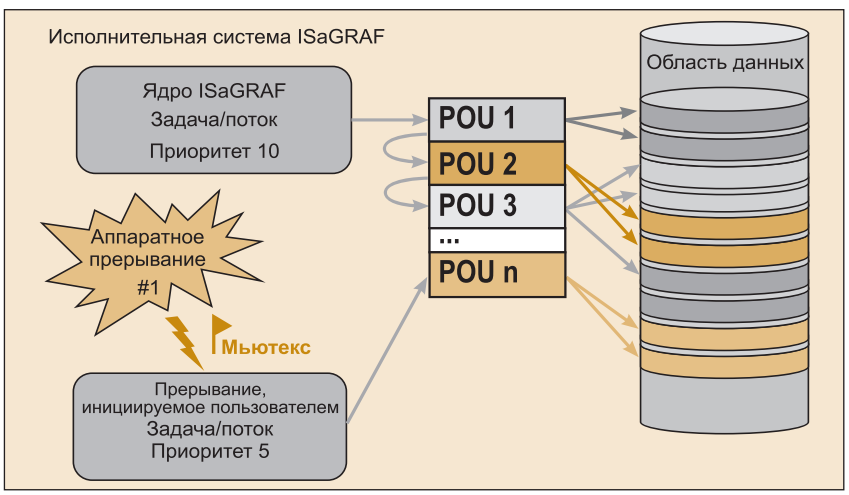

Рис.4

Узел прерывания позволяет пользователю получить доступ к интерфейсу прерывания (с помощью двойного клика на узле) и добавить новое программное прерывание (с помощью двойного клика и использования команды «Add»). Каждый POU, который создается как обработчик прерывания, пользователи могут связать с прерыванием, используя окна свойств. Плагин прерывания позволит пользователю управлять каждым прерыванием в любом ресурсе любого устройства.

Введение прерывания подразумевает возможность параллельного доступа к переменным (например, несколько прерываний, могут

одновременно запросить доступ к некоторой глобальной переменной). Для решения проблемы множественного доступа к данным были добавлены две новых функции работы с мьютексами: LOCK_CPU и UNLOCK_CPU. Эти функции имеют следующий синтаксис:

```
Bool:= LOCK_CPU (TIME);
Bool:= UNLOCK_CPU();
```

Рассмотрим пример: программа POU1 увеличивает счетчик, но при каждом прерывании программа POU2 должна сбросить его в 0. Программа POU1 содержит фрагмент кода:

```
LockState:= LOCK_CPU (T#500ms);
Counter := counter + 1;
LockState:= UNLOCK_CPU();
```

Программа POU2 содержит фрагмент кода:

```
LockState:= LOCK_CPU (T#500ms);
Counter := 0;
LockState:= UNLOCK_CPU();
```

Отметим, что на данный момент в режиме симуляции прерывания не поддерживаются. Эта возможность будет реализована в следующих версиях.

Расширение поддержки технологии ISaGRAF в области систем, связанных с безопасностью

В целях расширения поддержки технологии ISaGRAF в области систем, связанных с безопасностью, в декабре 2011 г. компания ISaGRAF Inc. анонсировала вывод на рынок платформы FlexiSafe на основе технологии ISaGRAF и стандартов IEC 61508 и ISO 13849. Некоторые из возможностей (Контроль версий, дерево зависимостей и др.) в ISaGRAF 6.1 как раз и создавались для поддержки решений, удовлетворяющих требованиям функционально безопасных систем. Платформа FlexiSafe предназначена для облегчения сертификации OEM-производителями средств промышленной автоматизации в соответствии со стандартами IEC 61508 («Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью») на уровне полноты безопасности SIL3 или ISO 13849 («Безопасность оборудования. Элементы систем управления, связанные с безопасностью») на уровне PLe. Платформа FlexiSafe ориентирована на разработку систем, которые поддерживают распределенные приложения, совмещающие безопасную и небезопасную функциональность, расширенное управление безопасностью и средства управления жизненным циклом приложений. FlexiSafe обеспечит основные элементы, необходимые для сертификации: сертификат SC3 (Systematic Capability уровня 3) стандарта IEC 61508 в редакции 2010 г., технологию встраиваемого ПО, которая может быть перенесена на любую «безопасную» ОС, включающую набор инструментов валидации и верификации, 100% тестовые отчеты по различным инструкциям TIC-кода, выполненные независимыми организациями, средства верификации кода приложения (разнообразные компиляторы), другие инструменты, помогающие конечным пользователем в сертификации функций безопасности и зависящие от концепции безопасности приложений (PLCopen Safety Function Blocks, Cause and Effect Diagram, Static Checker, Version Source Control, Cross-Reference Browser, Dependency Tree и др.)

В конце 2011 г. компания ISaGRAF Inc. стала участником программы Wind River Partner Validation Program. Компании Wind River и ISaGRAF Inc. будут сотрудничать на рынках энергетики, транспорта и управления процессами, предлагая решение для систем, связанных с безопасностью, состоящее из платформы Wind River VxWorks Cert и ISaGRAF FlexiSafe. Компания Wind River – первый поставщик сертифицированной ОС, который предлагает решение на уровне SIL3. В соответствии с соглашением ISaGRAF Inc. будет интегрировать и сертифицировать FlexiSafe в среде платформы Wind River VxWorks Cert, являющейся коммерческой ОС РВ для критически важных приложений, связанных с безопасностью, и которые должны быть сертифицированы по строгим требованиям IEC 61508 и другим стандартам ПО. В частности, комбинация платформы VxWorks Cert и решения ISaGRAF будет являться сертифицированной по уровню SIL3 стандарта IEC 61508, предлагая промышленным компаниям проверенные и испытанные решения, обеспечивая сокращение времени вывода на рынок их собственных продуктов, а также снижение расходов на разработку и техническое обслуживание.

Список литературы

1. *Золотарев С.В.* ISaGRAF 6: эволюция от среды программирования контроллеров к единой платформе автоматизации // Автоматизация в промышленности. №3. 2011
2. *Золотарев С.В., Кудрявцева М.Е.* О новых направлениях развития среды ISaGRAF: курс на поддержку функционально безопасных систем // Автоматизация в промышленности. №3. 2012
3. *Золотарев С.В.* Стандарты серии IEC 60870 на системы телемеханики и особенности реализации протокола IEC-60870-5-104 в среде программирования контроллеров ISAGRAF // Автоматизация и IT в энергетике, №2, 2010
4. *Колтунцев А.В., Золотарев С.В.* Стандарт IEC 61499 и система программирования контроллеров ISaGRAF 5: от теории к практике // Rational Enterprise Management. №2, 2009

Колтунцев Алексей Владимирович *– коммерческий директор,*
Золотарев Сергей Викторович *– канд. техн. наук, ведущий эксперт компании ФИОРД.*
Контактный телефон (812) 323-62-12.
E-mail: alex@fiord.com, zolotarev@fiord.com