ISaGRAF и некоторые современные подходы к реализации функционально безопасных систем

В статье обсуждаются некоторые подходы к реализации функционально безопасных систем с помощью технологии программирования контроллеров ISaGRAF. Теоретической основой обсуждаемой в статье проблематики является стандарт IEC 61508 («Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью») и спецификация безопасных функциональных блоков организации PLCOpen. Практические вопросы реализации предлагаемых в теории подходов рассматриваются применительно к среде программирования контроллеров ISaGRAF.

Компания «ФИОРД», г. Санкт-Петербург

Стандарты и спецификации в области функциональной безопасности систем

Важность решения проблемы создания функционально безопасных систем (другими словами, функционирование которых является безопасным, отсутствует неприемлемый риск применения), обеспечивающих защиту персонала, населения, окружающей среды, оборудования и продукции в случае «предназначенного использования» и в нештатных ситуациях, была поставлена на повестку дня в середине 70-х годов 20 века. После целого ряда крупных аварий на промышленных объектах в Европе и США [1], приведших к многочисленным жертвам среди людей, работы в этом направлении получили мощный толчок. В связи с проблемой функциональной безопасности в зарубежных источниках часто приводится пример аварии в 1974 году во Фликсборо (Великобритания) на химическом заводе компании «Нипро Кемикл Плант», на котором произошел мощный взрыв парового облака циклогексана с последующим крупным пожаром и гибелью 28 человек. Другой широко известный пример – катастрофа в итальянском городе Севезо в 1976 году, произошедшая в результате сбоя в производственном процессе на химическом предприятии швейцарской фирмы ICMESA, приведшая к выбросу в атмосферу диоксина и имевшая серьезные последствия для здоровья людей, животных и окружающей среды. Появившаяся в 1982 году «Директива Севезо» стала фундаментом современного законодательства в странах ЕЭС в области безопасности в промышленности и на транспорте. Для СССР и современной России примерами техногенных катастроф такого масштаба стали аварии на Чернобыльской АЭС и на Саяно-Шушенской ГЭС. Все эти аварии привели к осознанию того, что при разработке и эксплуатации систем требуется явным образом ответить на вопрос о том, какой риск несет для окружающих данная система, насколько серьезными могут быть последствия ее нештатной работы и что может быть сделано для уменьшения уровня риска. Сразу подчеркнем, что понятие функциональная безопасность не является синонимом термина надежность, хотя и тесно связано с ним. В некоторых случаях ненадежная система может быть абсолютно функционально безопасной. Например, частые сбои в работе пульта для телевизора не несут никакой прямой серьезной угрозы для жизни и здоровья человека (если, конечно, не учитывать ущерб психическому здоровью), и поэтому пульт для телевизора может считаться вполне функционально безопасным. И наоборот, очень надежная система (например, метро) является объектом повышенной опасности. Еще один важный момент: не следует путать между собой английские слова safety и security, которые на русский язык часто переводятся как «безопасность». Очень образно о разнице значений этих слов ска-

зано на одном из отечественных сайтов по изучению английского языка: «safety – это про технику безопасности, security – это про охрану порядка, safety officer – ответственный за соблюдение техники безопасности (например, на предприятии), security officer – сотрудник охраны (например, в банке)».

Прежде, чем перейти к рассмотрению конкретного технического вопроса о возможностях технологии ISaGRAF как инструмента создания программного обеспечения программируемых логических контроллеров (ПЛК), применяемых в приложениях, удовлетворяющих требованиям функциональной безопасности, остановимся подробнее на самом понятии функциональной безопасности. Термины «функциональная безопасность» (functional safety), «связанный с безопасностью» относятся к любым техническим и/или программируемым системам, отказ в которых, как одиночный, так и возникший в комбинации с другими отказами или ошибками, может привести к жертвам, травмам среди людей или к ущербу для окружающей среды. Термин функциональная безопасность соотносится с надежностью оборудования, обеспечивающего безопасность, и отражает вероятность правильного функционирования такого оборудования. Примерами систем, связанных с безопасностью, являются системы аварийного останова процесса, блокировки опасных механизмов, железнодорожная сигнализация, управление котлом и горелками, устройства обнаружения огня и утечек. Как видно, все приведенные системы обычно строятся с использованием ПЛК и поэтому задача создания «безопасных» ПЛК (то есть ПЛК, удовлетворяющих требованиям систем, связанных с безопасностью) является весьма актуальной. Очевидно, что эта задача для ПЛК требует адекватного решения как на аппаратном, так и на программном уровне, обеспечивая такую работу, что даже при наихудшем стечении обстоятельств отказ должен сказываться на процессе только предсказуемым, безопасным образом [2]. В данной статье мы не будем останавливаться на аппаратном уровне, а только вкратце упомянем некоторые принятые подходы к обеспечению необходимого уровня безопасности аппаратной части ПЛК: резервирование (которое позволяет поддерживать безопасность технологического процесса даже при отказе части оборудования), внутрисистемная аппаратная диагностика (которая позволяет аппаратно-программному комплексу с большой степенью достоверности диагностировать нештатную работу), дополнительные средства защиты операций чтения и записи по каналам связи.

Осознание необходимости формальных требований к функциональной безопасности привели к разработке базового стандарта в рамках Международной Электротехнической Комиссии: IEC 61508 – «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью». Соответствующая ему версия принята в России в качестве стандарта ГОСТ Р МЭК 61508. На базе и в дополнение к IEC 61508 принято ряд отраслевых и уточняющих стандартов, наиболее известными из которых являются IEC 61511 «Функциональная безопасность. Безопасность приборных систем для промышленных процессов», IEC 61513 «Атомные электростанции. Системы контроля и управления, важные для безопасности», IEC 62061 «Безопасность в машиностроении. Функциональная безопасность электронных и программируемых электронных систем управления машинами», IEC 60204-1 «Безопасность машин. Электрооборудование машин и механизмов», ISO 13849 «Безопасность оборудования. Элементы систем управления, связанные с безопасностью» (является развитием EN 954-1), EN 50216 «Объекты железнодорожного транспорта. Требования и подтверждение надежности, безотказности, обслуживаемости и безопасности». Кроме того, приведем названия еще некоторых стандартов Международной Организации по Стандартизации ISO, IEC и EN (Европейские нормы), на которые есть ссылки в данной статье: EN 574 «Безопасность машин. Приспособления двуручного управления», ISO 12100-1 «Безопасность машин. Основные понятия, общие принципы конструирования», EN 418 «Техника безопасности по машинам. Устройства аварийного отключения», IEC 61496-1 «Безопасность оборудования. Электрочувствительные защитные устройства. Часть 1: Общие требования и испытания», EN 953 «Безопасность машин. Ограждения. Общие требования к проектированию и конструированию неподвижных и перемещаемых ограждений», EN 1088 «Безопасность машин. Блокировочные устройства, связанные с защитными устройствами. Принципы конструирования и выбора», IEC 61800-5-2 «Системы силовых электрических приводов с регулируемой скоростью. Часть 5-2. Требования безопасности. Функциональная безопасность», IEE SEMSPLC «Прикладное программное обеспечение программируемых логических контроллеров, связанное с безопасностью».

В стандарте IEC 61508 предложено разделять системы на 4 уровня полноты безопасности SIL (Safety Integrity Level) в зависимости от назначения и требований к приемлемому фактору снижения риска возникновения опасного от-

Таблица 1. Уровни полноты безопасности SIL

Уровень полноты безопасности SIL	Назначение	Фактор снижения риска	Интенсивность опасных отказов при высокой интенсивности запросов (опасных отказов в час)	Интенсивность опасных отказов при низкой интенсивности запросов (вероятность отказа)
4	Защита от общей катастрофы	От 100,000 до 10,000	От 10–9 до 10–8	От 10–5 до 10–4
3	Защита обслуживающего персонала и населения	От 10,000 до 1,000	От 10–8 до 10–7	От 10–4 до 10–3
2	Защита оборудования и продукции; защита от травматизма	От 1000 до 100	От 10–7 до 10–6	От 10–3 до 10–2
1	Защита оборудования и продукции	От 100 до 10	От 10–6 до 10–5	От 10–2 до 10–1

каза (табл. 1) с учетом интенсивности таких отказов.

В стандарте IEC 61508 предлагается оценивать полноту безопасности двумя способами: количественно и качественно. Количественные методы в основном применимы для аппаратных средств и основаны на сравнении частоты случайных отказов аппаратуры с некоторой заданной величиной допустимого риска их появления и могут вызывать необходимость совершенствования проектных решений (например, путем введения дополнительного аппаратного резервирования). Качественные методы ориентированы на минимизацию так называемых «систематических» отказов (в том числе ошибок в программах) путем использования средств управления качеством, которые устанавливают требования и действия на всех стадиях жизненного цикла продукта, начиная от анализа возможных рисков, формулирования требований безопасности, и кончая стадией обслуживания законченного продукта, его эксплуатацией и утилизацией.

В стандарте IEC 61508 определено 16 этапов жизненного цикла продуктов, связанных с безопасностью. С целью уменьшения трудозатрат на разработку и сертификацию программного обеспечения (ПО) ПЛК организация PLCOpen (www.plcopen.org) предложила свой подход [3] для одного из этапов IEC 61508 «Этап 9: Реализация; Жизненный цикл программного обеспечения, связанного с безопасностью 9.1.1; Спецификация требований к функциям, связанным с безопасностью «. Таким образом, организация PLCOpen явным образом определила уровень своей компетенции в вопросе функциональной безопасности ПО ПЛК. Спецификация PLCOpen была реализована в рамках технологии программирования ISaGRAF [4], а также некоторыми другими поставщиками средств программирования контроллеров. PLCOpen рассматривает ПО ПЛК, связанного с безопасностью, в виде 2 уровней и выделяет адекватные им языки программирования: системное и прикладное программное обеспечение ПЛК, связанное с безопасностью:

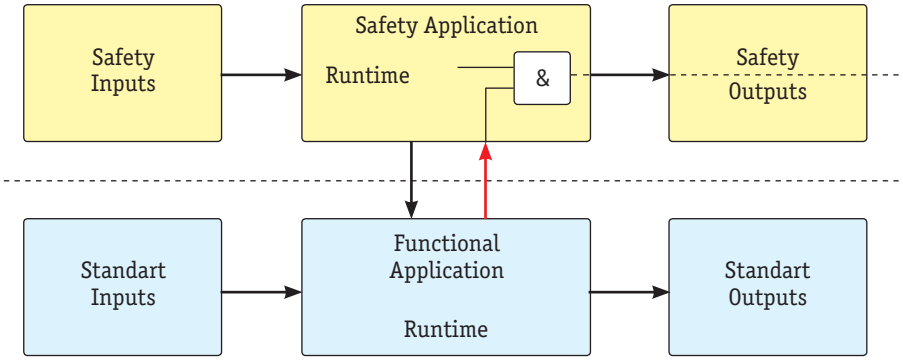

Рис. 1. Архитектурная модель PLCOpen для приложений на основе ПЛК, связанных с безопасностью

1. Системное программное обеспечение ПЛК, связанное с безопасностью: встраиваемое программное обеспечение, микропрограммы или операционная система, которая должно соответствовать положениям IEC 61508, особенно части 3. Языки, используемые здесь – C, C++, ассемблер. Это уровень языков с полной варьируемостью (Full Variability Languages): независимые от приложения языки поставщиков компонентов, используемые для реализации микропрограмм обеспечения функций безопасности, операционных систем или средств разработки. Они редко используются для реализации функций безопасности в прикладных программах. Эту задачу обычно решают поставщики системного ПО, такие как разработчики операционных систем VxWorks (компания Wind River, www.windriver.com), QNX Neutrino (компания QNX Software Systems, www.qnx.com), поставщики средств разработки VectorCAST (компания Vector Software, www.vectorcast.com). Обратим еще внимание на то, что для системного уровня приложений, связанных с безопасностью, как правило, требуется использование операционных систем реального времени (VxWorks, Integrity, QNX Neutrino,..), которые гарантируют детерминированное время реакции на события [5].

2. Прикладное программное обеспечение ПЛК, связанное с безопасностью. PLCopen предлагает использовать усеченные версии языков программирования, инструкции и сертифицированные функциональные блоки, например, ориентированные на целевые рынки в соответствии со стандартами IEC 62061 и ISO 13849-1. Это значительно упрощает разработку программного обеспечения и его сертификацию. Предложено применять языки с ограниченной варьируемостью (Limited Variability Languages). Они ориентированы на пользователей, которым требуется создавать приложения, связанные с безопасностью. Обычно для этого используются языки лестничных диаграмм (LD) и функциональных блоковых диаграмм (FBD). Заметим, что PLCOpen не рассматривает функциональные блоки в понимании стандарта IEC 62061, который допускает включение в рассмотрение аппаратных средств, обеспечивающих функциональность подсистемы безопасности.

PLCOpen вводит модель программного обеспечения для описания типовой взаимосвязи функциональных блоков, связанных с безопасностью, в рамках всей системы управления. Эта модель является достаточно общей и в ней исключены аппаратные средства обеспечения безопасности.

В предлагаемой модели архитектуры (рис. 1) проводится различие между прикладной (функциональной) частью приложения и частью приложения, связанного с безопасностью. Две части приложения могут выполняться на одном устройстве или они могут находиться на двух или более отдельных взаимосвязанных устройствах. Обмен данными между частями приложения, представленный пунктирной линией, может быть реализован по сети, через систему вывода-вывода

или разделяемую память в одном устройстве. Как правило, важным требованием является то, что не должно быть никакого нежелательного влияния на безопасность приложения от функциональной части приложения. В левой части модели определены два множества входов, в правой части – два уровня выходов. Посередине показаны две отдельные среды исполнения с соответствующими входами и выходами. Обмен разрешенными данными между подсистемой, связанной с безопасностью, и функциональной частью приложения отображается в центре. Как видно из рисунка,

- Функциональная (прикладная) часть приложения имеет доступ по чтению к «безопасным» входам и глобальным переменным (как указано с помощью левой стрелки).
- Небезопасные сигналы могут использоваться в приложении, связанном с безопасностью, однако только для управления выполнением прикладной части программы и не могут быть подключены непосредственно к безопасным выходам (как указывается с помощью правой стрелки и оператором AND).

В целях четкого разграничения между безопасными и стандартными сигналами предложено использовать специальное обозначение «безопасный» для данных, связанных с безопасностью. В частности, вводится специальный тип данных SAFEBOOL, применимый для входов и выходов внутри программного обеспечения. Выделение явным образом «безопасных» типов данных подразумевает признание программистом того, что сигналы влияют на безопасность системы и должны рассматриваться с особой осторожностью. Далее, исходя из этого предположения, связи между данными могут быть автоматически проверены на предмет выявления любых недопустимых связей между стандартными сигналами и «безопасными» сигналами. Хотя «безопасный» тип данных не может гарантировать, что действие сигнала является реально безопасным (например, в случае их неправильного подсоединения к периферии), однако, этот организационный инструмент может использоваться для сведения к минимуму ошибок в прикладных программах. Кроме того, такой подход упрощает и сокращает верификацию потока сигналов и сертификацию продукта. Возможными средствами поддержки безопасных типов могут быть различные средства отображения/представления безопасных типов данных и фактическая поддержка компилятором безопасных типов данных.

Новый тип «безопасных» данных SAFEBOOL – это тип данных, который применяется в средах, связанных с безопасностью, и представляет собой уровень более высокого уровня с точки зрения полноты (целостности) безопасности, который вводит различие между переменными, связанными с безопасностью и не связанными с ней. SAFEBOOL действует как BOOL в рамках системы, но может содержать дополнительную информацию (атрибуты), необходимую для задания состояния и уровня безопасности (например, может включать уровень производительности PL, полноту безопасности SIL, вероятность отказа при запросе PFD, вероятность отказа в час PFH). Такая информация может использоваться для вычисления SIL с помощью инструментов программирования. Существует, по крайней мере, два способа получения переменной SAFEBOOL на уровне приложений:

- Данные предоставляются самим устройством, операционной системы или микропрограммой. Они могут включать безопасную сеть.
- Данные предоставляются путем сочетания безопасных входов в самом приложении (например, как двухканальный безопасный вход).

Системы, связанные с безопасностью, должны основываться на «негативный» логике: безопасное значение SAFEBOOL должно быть по умолчанию FALSE. Разработчики приложений должны обеспечить, чтобы все переменные SAFEBOOL приводились по умолчанию к значению FALSE, а также в значение FALSE при инициализации и после любой ошибки.

Организация PLCOpen сформулировала следующие общие рекомендации и ограничения для программного обеспечения ПЛК, связанного с безопасностью:

Часть приложения, связанная с безопасностью, выполняется только как одна задача, однако прикладные (не связанные с безопасностью) части программы могут содержать несколько задач и выполняться на отдельном процессоре или устройстве.

- Часть программы, связанная с безопасностью, не должна прерываться прикладной частью приложения.
- После запуска цикла приложения, связанного с безопасностью, все значения соответствующих входных данных должны быть актуальными и стабильными в течение всего цикла.
- Выходы, связанные с безопасностью, не могут быть изменены самостоятельно только прикладной частью приложения.
- В программе, связанной с безопасностью, рекомендуется использовать сертифицированные функциональные блоки, например, как определено в спецификации PLCOpen. Пользователь, таким образом, может добиться более высокого уровня защиты (предотвращения) от ошибок.
- Функциональные блоки, связанные с безопасностью, должны применяться в языках FBD и LD стандарта IEC 61131-3, в то время как содержимое прикладной части приложения (в том числе функциональных блоков) может реализовываться на любом другом языке программирования (например, ST стандарта МЭК 61131-3, C) или даже в микропрограммах или аппаратном обеспечении.
- Каждый POU/FB в приложении, связанном с безопасностью, должен включать общедоступную информацию, содержащую следующее: автор, дата создания, дата выпуска, версия, история версий и функциональное описание (включая параметры вывода). Эта информация отображается как минимум во время сертификации, разработке программ и изменении программы. Доступ к этой информации может различаться в зависимости от типа использования, например, может быть частью функционального блока или может

ссылаться на другой источник (например, на веб-сервер).

Кроме общих (независящих от уровня приложения) рекомендаций PLCOpen выделяет три уровня приложений и соответствующих им рекомендаций и ограничений с точки зрения сертификации на соответствие с требованиями функциональной безопасности: базовый, расширенный и системный уровень. Для базового уровня фундаментальный подход заключается в том, что программа, связанная с безопасностью, состоит только из сертифицированных функциональных блоков, которые могут «соединяться» (“wired”) друг с другом в графической форме. Эти программы имеют четкую структуру и могут легко читаться. Кроме того, время вывода на рынок таких программ значительно сокращается, так как они состоят из блоков, сертифицированных заранее. Расширенный уровень требуется в случае осуществления проектов, для которых текущий набор сертифицированных функциональных блоков недостаточен и пользователь может сам создавать требуемые блоки (или даже программы) на расширенном уровне. Для этого предоставляется диапазон расширенных команд. Однако валидация функциональности для этих блоков и программ может быть значительно более сложной и поэтому требуется больше времени, так как в основе этого лежит трудоемкая сертификация всего процесса. Системный уровень должен использоваться поставщиками систем контроля безопасности. Системный уровень реализован в специальных языках, однако он не является частью спецификации PLCOpen.

IEC 61508, часть 7, определяет ограничения в языках программирования для различных уровней полноты безопасности SIL (в терминах «Весьма рекомендованные», «Рекомендованные» или «Не рекомендованные»). Основываясь на этом, рекомендованными языками в спецификации PLCOpen являются графический язык функциональных блоковых диаграмм (FBD) и лестничные диаграммы (LD). Эти графические языки обеспечивают четкое представление о самой программе, связанной с безопасностью, и инструментальные средства с их использованием могут обеспечивать гораздо более высокий уровень поддержки и сопровождения для пользователей. Они формирует основу для упрощенной эксплуатации программ, связанных с безопасностью. Структурированный текст (ST), список инструкций (IL) и язык последовательных функциональных схем (SFC) не рассматриваются PLCOpen на данный момент, так как они требуют более высоких затрат на поддержку в процессе жизненного цикла. Говоря более конкретно, тестирование и валидация приложений, написанных на ST или IL, является более сложной, чем для приложений на графических языках. Эта рекомендация относится к базовому и расширенному уровню. Для системного уровня (см. IEC 61508, часть 7) нет никаких рекомендаций для языков, функций и типов данных. PLCOpen предлагает ввести ограничения на типы типов данных в зависимости от уровня приложения для базового и расширенного уровней. В таблицах в документе [3] значение «X» указывает на то, что элемент разрешен, «-» указывает на то, что элемент не разрешен. Типы

Таблица 2. Функциональные блоки, связанные с безопасностью, реализованные в ISaGRAF

Функциональный блок	Описание	Ссылочные стандарты
SF_Antivalent	Преобразование двух безопасных несовпадающих входов в один безопасный выход с контролем времени несовпадения	EN 954-1
SF_EDM	Управление безопасными выходами и мониторинг контролируемых исполнительных механизмов: контролируемая остановка с сохранением подвода питания к исполнительным механизмам	IEC 60204-1, EN 954-1, ISO 12100-2
SF_EmergencyStop	Мониторинг кнопки аварийной остановки и запуска аварийной остановки	EN 418, EN 954-1, ISO 12100-2, EN 60204-1
SF_EnableSwitch	Оценка сигналов от трехпозиционного переключателя	IEC 60204-1, EN 954-1, ISO 12100-2
SF_Equivalent	Преобразование двух безопасных эквивалентных входов в один безопасный выход с контролем времени несовпадения	EN 954-1
SF_ESPE	Мониторинг электрочувствительного защитного устройства	IEC 61496-1, EN 954-1, ISO 12100-2
SF_GuardLocking	Управление блокирующим защитным ограждением с 4 состояниями защитного ограждения («блокирующее защитное ограждение с фиксацией закрытия»)	EN 953, EN 1088, EN 954-1, ISO 12100-2:
SF_GuardMonitoring	Мониторинг блокирующего защитного ограждения с двумя переключателями и контролем времени	EN 953, EN 1088, EN 954-1, ISO 12100-2
SF_ModeSelector	Выбор режима работы системы (ручного, автоматического, полуавтоматического,..)	MRL 98/37/EC, Annex I, EN ISO 12100-2, IEC 60204-1, EN 954-1, ISO 12100-2
SF_MutingPar	Управление функциями безопасности с использованием параллельного отключения с четырьмя датчиками отключения	IEC 61496–1, IEC 62046, EN 954-1, ISO 12100–2
SF_MutingPar_2Sensor	Управление функциями безопасности с использованием параллельного отключения с двумя датчиками отключения	IEC 61496-1, IEC 62046, EN 954-1, ISO 12100-2
SF_MutingSeq	Управление функциями безопасности, с использованием последовательного отключения с четырьмя датчиками отключения	IEC 61496-1, IEC 62046, EN 954-1, ISO 12100-2
SF_OutControl	Контроль безопасного выхода с помощью сигналов безопасности и сигнала прикладной части приложения	EN 954-1, ISO 12100-2, EN 60204-1
SF_SafelyLimitedSpeed	Активация мониторинга безопасного уменьшения скорости	IEC 61800-5-2, EN 954-1, ISO 12100-2, EN 60204-1
SF_SafeStop1	Начало контролируемой остановки (IEC 60204–1, Категория 1 – контролируемая остановка с сохранением подвода питания к исполнительным механизмам до самой остановки машины, с последующим отключением подвода питания после того, как остановка осуществлена.)	IEC 61800–5-2, EN 954–1, ISO 12100–2, EN 60204–1
SF_SafeStop2	Начало контролируемой остановки (IEC 60204–1, Категория 2 – контролируемая остановка с сохранением подвода питания к исполнительным механизмам.)	IEC 61800-5-2, EN 954-1, ISO 12100-2, EN 60204-1
SF_SafetyRequest	Установка привода в безопасное состояние	EN 954-1, ISO 12100-2, EN 60204-1
SF_TestableSafetySensor	Обнаружение сигналов потери значения датчика, превышения времени отклика или статического «ВКЛ»	IEC 61496-1, EN 954-1, ISO 12100-2
SF_TwoHandControlTypeII	Функциональность двуручного управляющего устройства (EN 574, Section 4, Type II). Тип II требует: наличия двух приборов управления исполнительными механизмами для согласованного воздействия двумя руками, удерживающего воздействия в присутствии опасных ситуаций, прерывание работы, если один из органов управления отпущен, в присутствии опасной ситуация, освобождения обоих органов управления исполнительными механизмами перед повторным запуском.	EN 574, ISO 12100-2
SF_TwoHandControlTypeIII	Функциональность двуручного управляющего устройства (EN 574, Section 4, Type III). Тип III: то же, что и Тип II плюс приводиться в действие в ограниченный промежуток времени, не превышающий 0,5 с, и, если это предельное время превышено, оба устройства управления должны быть отпущены перед тем. как появится возможность нового запуска.	EN 574, ISO 12100-2

данных, отличные от SAFEBOOL, могут кроме того иметь атрибут «safe», например, SAFEINT, для того чтобы разрешить автоматически отслеживать безопасные данные. Мы не будем приводить здесь эти таблицы: интересующиеся могут самостоятельно ознакомиться с ними, скачав (после регистрации) нужные документы с сайта PLCOpen http://www.plcopen.org/pages/tc5_safety/specifications/.

В целях детализации возможностей базового уровня приложений организацией PLCOpen было предложено 19 «безопасных» функциональных блоков (таблица 13 в документе [3]), которые в полном объеме реализованы в ISaGRAF. Для каждого из «безопасных» функциоанльного блока (ФБ) в спецификации PLCOpen приводится подробное и краткое описание интерфейса, перечень ссылочных стандартов, описание функционирования в текстовой и графической форме (детальный граф состояний), временные диаграммы, включая стадии нормальной работы и поведения в начальной стадии, описание ошибок и способы их выявления, работу ФБ при возникновении ошибок, коды состояний и диагностики ФБ. В ISaGRAF реализованы все 19 функциональных блоков, предложенных PLCOpen и связанных с безопасностью (табл. 2).

Кроме того, в ISaGRAF полностью реализована концепция диагностических кодов DiagCode для всех функциональных блоков, связанных с безопасностью. Значения DiagCode образуют единую систему диагностики вне зависимости от поставщика. Кроме того, разрешено добавлять собственные дополнительные сведения в значение выхода DiagCode. В качестве примера приведем значение DiagCode 8002hex, которое означает, что функциональный блок активирован, зафиксирован запрос к функции безопасности и в результате выполнения ФБ безопасный выход будет установлен в значение TRUE.

Новые направления развития технологии ISaGRAF в области систем, связанных с безопасностью

В целях расширения поддержки технологии ISaGRAF в области систем, связанных с безопасностью, в декабре 2011 года компания ISaGRAF Inc. анонсировала вывод на рынок платформы FlexiSafe на основе технологии ISaGRAF и стандартов IEC 61508 и ISO 13849. Платформа FlexiSafe предназначена для облегчения сертификации OEM-производителями средств промышленной автоматизации в соответствии со стандартами IEC 61508 на уровне SIL3 или ISO 13849 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. С более подробной информацией о FlexiSafe можно ознакомиться в буклете «ISaGRAF FlexiSafe: IEC 61508 SC3, ISO 13849 PLe» (http://www.fiord.com/images/industry_avt/soft/isagraf/FlexiSafe-flyer.pdf). FlexiSafe обеспечит основные элементы, необходимые для сертификации: сертификат SC3 (Systematic Capability уровня 3) стандарта IEC 61508 в редакции 2010 года, технологию встраиваемого программного обеспечения, которая может быть перенесена на любую «безопасную» (safety) операционную систему, 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 сертификации конечным пользователем функций безопасности, зависящие от концепции безопасности приложений (PLCopen Safety Function Blocks, Cause and Effect Diagram, Static Checker, Version Source Control, Cross-Reference Browser, Dependency Tree,..).

С точки зрения безопасности, платформа FlexiSafe позволяет многократно использовать результаты сертификации применительно к различным аппаратным платформам, упрощает разработку и сертификацию приложений конечного пользователя, допускает сертификацию резервированных конфигураций. FlexiSafe предлагает следующий подход к сертификации:

- Портирование сертифицированной исполнительной среды (runtime) на целевую аппаратную платформу и операционную систему (принимая во внимание FlexiSafe и руководство по безопасности ОС).
- Некоторые средства оценки мер безопасности, включенные в исполнительную среду.
- Сервисы проверки портирования, использующие «строгий» (rigourous) системный слой и всеобъемлющий набор тестового покрытия.
- Не требуется 100% тестирование приложения конечного пользователя (только функциональные тесты).
- Приложения конечного пользователя могут использовать преимущества всех языков стандарта IEC 61131-3, включая SFC.
- Никаких ограничений на использование функциональных блоков.
- Интегрированное управление безопасностью и средствами жизненного цикла приложений, соответствующее стандартам IEC 61508 и ISO 13849.

С точки зрения требований стандарта IEC 61508, верификация и валидация ядра ISaGRAF в рамках FlexiSafe основана на допустимом в IEC 61508 подходе – «доказано практикой» («Proven-in-Use»). ISaGRAF используется в условиях реальной эксплуатации в течение 14 лет в составе 850 000 исполнительных систем в ответственных промышленных приложениях, требующих обеспечения безопасности. Такие данные по результатам эксплуатации ISaGRAF позволяют сделать заключение о превышении требований к уровню SIL3 (смотрите таблицу D.1 «Необходимая предыстория для определения уровня полноты безопасности» в части 7 стандарта IEC 61508): для SIL3 общее число часов эксплуатации должно превышать 3×10^8 (при доверительной вероятности 0.95).

В ISaGRAF обеспечена инкапсуляция функций безопасности: и оболочка ISaGRAF строится вокруг

функциональности, которая проверяет правильность и безопасность функционирования. Портируемое тестовое окружение ISaGRAF поддерживает корректную функциональность и отсутствие побочных эффектов, сочетает Black-box и White-box испытания; покрывающие все операции ядра (I/O, обновления, и т.д.) и каждую инструкцию TIC-кода. Для подтверждения этого факта предоставляются документально зафиксированные результаты (отчеты) динамического тестирования. При проведении сертификации ISaGRAF были использованы комплексные инструментальные средства VectorCAST компании Vector Software (www.vectorcast.com). которые значительно снижают время, усилия и затраты, связанные с тестированием компонентов программного обеспечения, необходимых для проверки безопасности встраиваемых систем.

Упомянем еще тот факт, что компания ISaGRAF Inc. в конце 2011 года стала участником программы Wind River Partner Validation Program. Компании Wind River и ISaGRAF Inc. будут сотрудничать на рынках энергетики, транспорта и управления процессами с помощью решения для систем, связанных с безопасностью, и состоящего из платформы Wind River VxWorks Cert и ISaGRAF FlexiSafe. Wind River – это ведущий поставщик ОСРВ и первый поставщик сертифицированной ОСРВ, который предложил решение на уровне SIL3. В соответствии с соглашением ISaGRAF Inc. будет интегрировать и сертифицировать FlexiSafe в среде платформы Wind River VxWorks Cert, являющейся коммерческой операционной системой реального времени для критически важных приложений, связанных с безопасностью, и которые должны быть сертифицированы по строгим требованиям IEC 61508 и другим стандартам программного обеспечения. В частности, комбинация платформы VxWorks Cert и решения ISaGRAF будет являться сертифицированным 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. А расходы эти могут быть весьма значительными, особенно, если каждый раз выполнять сертификацию «с нуля». Например, известны данные [5] по аналогичному по трудоемкости процессу сертификации по стандарту DO-178B: около 125 строк кода за один человеко-месяц. Поэтому снижение затрат (финансовых, трудовых, временных) на сертификацию решений на основе ISaGRAF может быть весьма значительным и сыграть решающую роль.

Литература

1. Дэвид Дж. Смит, Кеннет Дж. Л. Симпсон. Функциональная безопасность. Простое руководство по применению стандарта МЭК 61508 и связанных с ним стандартов, Москва, Издательский дом «Технологии», 2004

2. Dr. William M. Goble, Conventional PLC vs. Safety PLC. Controllers for Safety Instrumented Systems, http://www.exida.com/articles/cvsplc_rev1.pdf

3. PLCopen – Technical Committee 5. Safety Software.Technical Specification. Part 1: Concepts and Function Blocks, 2006

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 г.

5. Золотарев С. В., Современные операционные системы реального времени для перспективной авионики, Военный парад, № 6, 2006 г.

С. В. Золотарев, к. т.н., ведущий эксперт,
М. Е. Кудрявцева, аспирантка СПбГУ, менеджер направления программных средств,
Компания «ФИОРД», г. Санкт-Петербург,
тел.: (812) 323-6212,
e-mail: info@fiord.com,
www.fiord.com