С. В. Золотарёв, компания «ФИОРД»

ISaGRAF и функционально безопасные системы: современный тренд в развитии технологий программирования контроллеров

В статье обсуждается один из современных трендов в развитии технологий программирования контроллеров – поддержка функционально безопасных систем (safety system, «систем, связанных с безопасностью») в понимании стандарта IEC 61508 («Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью») и спецификации организации PLCOpen безопасных функциональных блоков программного обеспечения ПЛК. Некоторые практические вопросы реализации функционально безопасных систем рассматриваются применительно к технологии программирования контроллеров ISaGRAF и платформы FlexiSafe, ориентированной на выполнение задач по сертификации OEM-производителями средств промышленной автоматизации в соответствии со стандартами IEC 61508 на уровне SIL3 или ISO 13849 на уровне производительности PLe.

Функционально безопасные системы: актуальность, стандарты и спецификации

Обсуждение проблематики функционально безопасных систем хотелось бы начать с замечания о том, что абсолютно безопасных систем не существует. После принятия необходимых защитных мер всегда имеется ненулевая вероятность остаточного риска, и безопасность достигается путём его уменьшения до приемлемого уровня, определённого как допустимый (остаточный) риск применения. Важность решения проблемы создания функционально безопасных систем, 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 серьёзный толчок. В контексте обсуждения проблемы функциональной безопасности 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 серьёзные 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 формализованным образом ответить на вопрос о том, какой риск несёт для окружающей среды данная система, насколько серьёзными могут быть последствия её нештатной работы и что может быть сделано для уменьшения уровня риска. Понимание необходимости формализованных требований привело к разработке в рамках Международной

Электротехнической Комиссии базового стандарта 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. Заметим, что следование этому стандарту при производстве продукции и реализации проектов – это весьма серьёзный шаг, требующий значительных финансовых, трудовых, временных и интеллектуальных затрат, но в данном случае, конечно, цель оправдывает средства.

Подчеркнём, что понятие функциональная безопасность не является синонимом термина надёжность, хотя и тесно связано с ним. В некоторых случаях ненадёжная система может быть абсолютно функционально безопасной. Например, частые сбои в работе пульта для телевизора не несут никакой прямой серьёзной 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 надёжная система (например, метро) является объектом повышенной опасности. Ещё 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 сказано 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)». Раз уже разговор у нас идёт о терминах, остановимся на терминах «валидация» и «верификация», которые мы будем употреблять в статье. Весьма доходчиво о них сказано в книге [5]: «валидация» отвечает на вопрос «Правильная ли выполняется работа?», «верификация» отвечает на вопрос «Правильно ли выполняется работа?». Казалось бы, разница всего в двух буквах, но смысл и содержание понятий совершенно разный.

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 надёжностью оборудования, обеспечивающего безопасность, и отражает вероятность правильного функционирования такого оборудования. Типовые примеры объектов и процессов, связанных с безопасностью, приведены на сайте IEC (http://www.iec.ch/functionalsafety/faq-ed1/): аварийное отключение, пожарные и газовые системы, турбины, управление газовой горелкой, защита и блокировка при экстренной остановке машины, медицинские приборы, динамическое позиционирование (контроль движения судна), железнодорожные сигнальные системы, изменение скорости вращения двигателя (как средство защиты), удалённый мониторинг, эксплуатация и программирование распределённых технологических процессов и другие объекты и процессы. Как видно, многие упомянутые системы обычно строятся с использованием ПЛК, и поэтому задача создания «безопасных» ПЛК (то есть ПЛК, удовлетворяющих требованиям систем, связанных с безопасностью) весьма актуальна. 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. 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. За рамками данной статьи останутся также вопросы, связанные с проблематикой защиты от целенаправленных атак на средства АСУ ТП (SCADA-пакеты и ПЛК).

На базе стандарта IEC 61508 и в дополнение к нему приняты ряд отраслевых и уточняющих стандартов, наиболее известные из которых – 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 надёжности, безотказности, обслуживаемости и безопасности». Кроме того, приведём названия ещё 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 отказа (таблица 1) с учётом 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, путём введения дополнительного аппаратного резервирования). Качественные методы ориентированы на минимизацию так называемых «систематических» отказов (в том числе ошибок в программах) путём 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. Это «Этап 9: Реализация; Жизненный цикл программного обеспечения, связанного с безопасностью 9.1.1; Спецификация требований к функциям, связанным с безопасностью». Таким образом, организация PLCOpen в явном виде 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. Организация PLCOpen рассматривает ПО ПЛК, связанного с безопасностью, в виде 2-х уровней и выделяет адекватные им языки программирования. Эти 2 уровня – связанное с безопасностью системное и прикладное программное обеспечение ПЛК.

1. *Системное ПО ПЛК, связанное с безопасностью*: встраиваемое ПО, микропрограммы или ОС. Используемые языки: C, C++, ассемблер. Это уровень языков с полной вариативностью: независимые от приложения языки, используемые для реализации микропрограмм функций безопасности, операционных систем или средств разработки. Эти языки используются обычно поставщиками системного ПО и редко для реализации функций безопасности в прикладных программах. Обратим ещё внимание на то, что для системного уровня приложений, свя-

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 100000 до 10000	От 10^{-9} до 10^{-8}	От 10^{-5} до 10^{-4}
3	Защита обслуживающего персонала и населения	От 10000 до 1000	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}

занных с безопасностью, как правило, требуется использование ОС РВ (VxWorks, Integrity, QNX Neutrino и др.), которые гарантируют детерминированное время реакции на события. В спецификации PLCOpen системный уровень детально не рассматривается, а лишь выделяется.

2. *Прикладное ПО ПЛК («приложение»), связанное с безопасностью.* Организация PLCopen предлагает использовать на данном уровне усечённые версии языков программирования (с ограниченной вариативностью), инструкции и сертифицированные функциональные блоки (ФБ), например ориентированные на целевые рынки в соответствии со стандартами IEC 62061 и ISO 13849-1. Предложено применять языки с ограниченной вариативностью, например языки лестничных диаграмм (LD) и функциональных блоковых диаграмм (FBD). Это значительно упрощает разработку ПО и его сертификацию. Заметим, что PLCOpen не допускает включение в рассмотрение аппаратные средства, обеспечивающие функциональность подсистемы безопасности.

Модель прикладного ПО для систем, связанных с безопасностью

Организация PLCopen предлагает собственную модель прикладного ПО для систем, связанных с безопасностью, которая базируется на описании типовой взаимосвязи функциональных блоков (ФБ). В предлагаемой модели архитектуры (рис. 1) предлагается разделять прикладное ПО ПЛК («приложение») на функциональную часть приложения и часть приложения, связанную с безопасностью. Эти части приложения могут выполняться на одном вычислительном устройстве или на двух или более отдельных, но взаимосвязанных устройствах. Обмен данными между этими частями приложения, представленный пунктирной линией, может быть реализован по сети, через систему вывода-вывода или разделяемую память, если приложения выполняются на одном устройстве. Важным требованием является отсутствие любого нежелательного влияния на часть приложения, связанную с безопасностью, со стороны функциональной части приложения. Исходя из этого требования на рис. 1, в левой части модели изображены два множества входов (стандартные и безопасные), в правой части – два уровня выходов (стандартные и безопасные). Средние блоки символизируют две отдельные среды исполнения. Обмен разрешёнными данными между частью приложения, связанной с безопасностью, и функциональной частью приложения отображается стрелками. Функциональная часть приложения имеет доступ к безопасным входам только по чтению (вертикальная стрелка 1 на рис. 1) и глобальным переменным. Небезопасные сигналы входа не могут быть подключены непосредственно к безопасным выходам (вертикальная стрелка 2 и оператор AND на рис. 1), однако они могут также подаваться на вход ФБ в части приложения, связанной с безопасностью (на рис. 1 это не показано).

В целях чёткого разграничения между безопасными и стандартными сигналами организацией PLCOpen предложено использовать для данных, связанных с безопасностью, специальный префикс SAFE – «безопасный». В частности, вводится специальный тип данных SAFEBOOL, применимый для входов/выходов внутри ПО и используемый только в части приложения, связанной с безопасностью. Выделение явным образом «безопасных» типов данных подразумевает признание того, что сигналы влияют на безопасность системы и должны рассматриваться с особой осторожностью. Исходя из этого предположения, связи между данными могут быть автоматически проверены на предмет выявления любых недопустимых связей между стандартными и безопасными сигналами. Хотя «безопасный» тип данных не может гарантировать, что действие сигнала реально безопасно (например, в случае их неправильного подсоединения к периферии), однако этот организационный инструмент может использоваться для сведения к минимуму ошибок в приложениях. Кроме того, такой подход упрощает и сокращает верификацию потока сигналов и сертификацию продукта. Возможными средствами поддержки «безопасных» типов данных могут быть различные средства их отображения/представления (например, специальным цветом или специальным типом данных) и фактическая поддержка компилятором.

Новый тип «безопасных» данных SAFEBOOL– это тип данных, который применяется в средах, связанных с безопасностью. Тип SAFEBOOL представляет собой более высокий уровень с точки зрения полно-

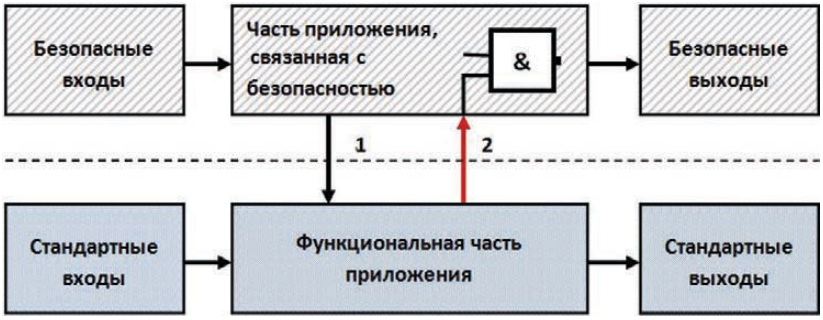

Рис. 1. Архитектурная модель ПО для систем, связанных с безопасностью

ты (целостности) безопасности и вводит различие между переменными, связанными и не связанными с безопасностью. Тип 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, вероятность отказа при запросе PFD, вероятность отказа в час PFH). Такая информация может использоваться для вычисления полноты безопасности SIL с помощью инструментов программирования. Существует по крайней мере два способа формирования переменной SAFEBOOL на уровне приложений:

1) данные предоставляются самим устройством, ОС или микропрограммой;

2) данные предоставляются путём 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 ПО ПЛК, связанного с безопасностью.

- Часть приложения, связанная с безопасностью, выполняется как единая задача, однако функциональная (не связанная с безопасностью) часть приложения может состоять из нескольких задач и выполняться на отдельном процессоре или устройстве.
- Часть приложения, связанная с безопасностью, не должна прерываться функциональной 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 напрямую изменены функциональной 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 реализовываться на языках FBD и LD стандарта IEC 61131-3, в то время как содержимое функциональной 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 (Program Organization Unit/ФБ) в части приложения, связанной с безопасностью, должен включать общедоступную информацию, содержащую следующие сведения: автора, дату создания и выпуска, версию, историю версий и функциональное описание (включая параметры вывода). Эта информация отображается во время сертификации, разработки и изменении программы и может быть частью функционального блока или указываться, например, в виде гиперссылки.

Кроме общих рекомендаций, относящихся ко всем программным приложениям, PLCOpen выделяет три уровня приложений и соответствующие им рекомендации и ограничения с точки зрения сертификации на соответствие требованиям функциональной безопасности: базовый, расширенный и системный. Для базового уровня фундаментальный подход заключается в том, что программа, связанная с безопасностью, состоит только из сертифицированных функциональных блоков, которые могут «соединяться» друг с другом в графической форме. Программы, составленные из этих функциональных блоков, имеют чёткую 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). Для этого ему предоставляется диапазон расширенных команд. Однако валидация функциональности для этих блоков и программ может быть значительно более сложной и потребует больше времени, так как в основе этого лежит трудоёмкая сертификация всего процесса. Системный уровень должен использоваться поставщиками средств контроля безопасности. Системный уровень, как правило, реализуется специальными средствами на базе операционной системой реального времени (ОСРВ), однако он не является частью спецификации PLCOpen (а лишь выделяется, как отдельный уровень).

Стандарт IEC 61508, часть 7, определяет ограничения в предпочтительных языках программирования для различных уровней полноты безопасности SIL (в терминах «Весьма рекомендованные» (HR), «Рекомендованные» (R) или «Нерекомендованные» (NR)). Основываясь на этом, рекомендованными языками в спецификации PLCOpen являются графический язык

функциональных блоковых диаграмм (FBD) и лестничные диаграммы (LD). Эти графические языки обеспечивают чёткое 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 формируют 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, – задача более сложная, чем для приложений на графических языках. Эта рекомендация относится к базовому и расширенному уровню. Для системного уровня рекомендации для языков, функций и типов данных отсутствуют.

PLCOpen предлагает ввести ограничения на типы данных в зависимости от уровня приложения для базового и расширенного уровней. В таблицах в документе [3] значение «X» указывает на то, что элемент разрешён, «–» указывает на то, что элемент не разрешён. Типы данных, отличные от SAFEBOOL, могут кроме того иметь атрибут «safe», например SAFEINT, для того чтобы разрешить автоматически отслеживать безопасные данные.

В целях детализации возможностей базового уровня приложений организацией PLCOpen было предложено 19 «безопасных» функциональных блоков (таблица 13 в документе [3]), которые в полном объёме реализованы в ISaGRAF. Для каждого из «безопасных» функциональных блоков (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 (таблица 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. Разрешено также добавлять собственные дополнительные сведения в значение выхода DiagCode. В качестве примера приведём значение DiagCode 8002_{hex}, которое означает, что функциональный блок активирован, зафиксирован запрос к функции безопасности и в результате выполнения ФБ безопасный выход будет установлен в значение TRUE.

Новый тренд в развитии ISaGRAF: платформа FlexiSafe для поддержки 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 распределённые 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). Платформа 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, с набором инструментов валидации и верификации;
- 100-процентные тестовые отчёты 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. Подход

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 трёхпозиционного 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 четырьмя 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, SO 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.	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 приводится 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

к сертификации, предлагаемый платформой FlexiSafe, выглядит следующим образом:

– производится портирование сертифицированной исполнительной среды (runtime) на целевую аппаратную платформу и операционную систему (принимая во внимание FlexiSafe и руководство по безопасности ОС);
– используются некоторые средства оценки мер безопасности, включённые в исполнительную среду;
– для проверки портирования предусмотрены сервисы, использующие «строгий» (rigourous) системный слой и всеобъемлющий набор тестового покрытия;
– не требуется 100-процентное 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»). Технология ISaGRAF используется в условиях реальной эксплуатации в составе 850000 исполнительных систем в ответственных промышленных приложениях, требующих обеспечения безопасности, уже 14 лет. 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 – 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 (ввод-вывод, обновления и т .д.) и каждую инструкцию TIC-кода. Для подтверждения этого факта предоставляются документально зафиксированные результаты (отчёты) 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. На основе анализа рисков в отношении исходного кода встраиваемых систем рекомендуется выполнение показанных на рис. 2 видов работ.

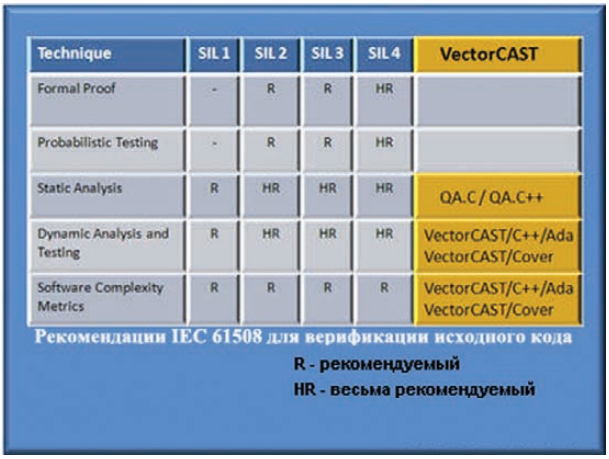

Рис.2. Рекомендации стандарта IEC 61508 для верификации исходного кода

Упомянем ещё 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, которое состоит из платформы Wind River VxWorks Cert и ISaGRAF FlexiSafe. В соответствии с соглашением, *ISaGRAF Inc.* будет интегрировать и сертифицировать FlexiSafe в среде платформы Wind River VxWorks Cert, являющейся коммерческой ОСРВ для критически важных приложений, которые связаны с безопасностью и должны быть сертифицированы по строгим требованиям IEC 61508 и другим стандартам программного обеспечения. 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. Ещё раз подчеркнём, что расходы на сертификацию продукции могут быть весьма значительными, особенно если каждый раз выполнять сертификацию «с нуля». Например, известны данные по аналогичному по трудоёмкости 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.

В заключение обратим внимание на последнюю (на момент написания статьи) версию ISaGRAF 6.1, в которую вошли многие новые средства (контроль версий, дерево зависимостей...). Эти средства необходимы для поддержки части приложения, связанного с безопасностью. Контроль версий исходных кодов (Version Source Control) обеспечивает поддержку совместной работы нескольких пользователей над одними и теми же элементами (например, такими как устройство, ресурс, программный модуль). Это также даёт 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. Дерево зависимостей (Dependency Tree) даёт 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. Исходя из всего выше изложенного, можно говорить о том, что ISaGRAF 6.1 и FlexSiafe отражают современный тренд в развитии технологий программирования контроллеров, который также активно поддерживается другими ведущими поставщиками в этой области (*KW-Software*, *Infoteam*, ...).

Список литературы

1. Дэвид Дж.Смит, Кеннет ДжЛ.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., Золотарёв С.В., Стандарт IEC 61499 и система программирования контроллеров ISaGRAF 5: от теории к практике, Rational Enterprise Management, №2, 2009 г.

5. Сергей Зыль. Проектирование, разработка и анализ программного обеспечения систем реального времени. Санкт-Петербург, 2010, «БХВ-Петербург».

Об авторе

Золотарёв Сергей Викторович – канд. техн. наук, ведущий эксперт компании «ФИОРД».

Телефон: (812) 323-62-12.

E-mail: info@fiord.com

http://www.fiord.com,

www.isagraf.ru, www.fit-pc.ru