

УДК 004.021

Н.А. Долбня, К.В. Ларин, В.В. Шишкин

МЕТОДИКА СОЗДАНИЯ ВИРТУАЛЬНОГО СТЕНДА АВИАЦИОННОЙ БОРТОВОЙ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ

Долбня Николай Алексеевич, аспирант, окончил механико-математический факультет Московского государственного университета им. М.В. Ломоносова. Начальник тематической конструкторской бригады ОАО «УКБП». Имеет публикации в области автоматизации проектирования бортовых информационных систем. [e-mail:dolbnya_na@ukbp.ru].

Ларин Кирилл Валентинович, кандидат технических наук, окончил инженерно-физический факультет Ленинградского института точной механики и оптики. Главный конструктор ОАО «УКБП». Имеет публикации в области автоматизации проектирования бортовых информационных систем. [e-mail:lucky@ukbp.ru].

Шишкин Вадим Викторович, кандидат технических наук, окончил радиотехнический факультет Ульяновского политехнического института. Профессор кафедры «Измерительно-вычислительные комплексы» УлГТУ. Имеет статьи в области разработки САПР, интеллектуальных систем, операционных систем реального времени и встраиваемых систем. [e-mail:shvv@ulstu.ru].

Аннотация

В статье описывается методика создания виртуального стенда авиационной бортовой информационно-управляющей системы (БИУС). Виртуальный стенд разворачивается в качестве клиент-серверного приложения в локальной вычислительной сети и позволяет отлаживать функциональное программное обеспечение БИУС без бортового вычислителя. При этом физический уровень каналов обмена системы заменяется транспортным уровнем протокола TCP. Методика также позволяет разрабатывать и отлаживать диагностическое обеспечение системы. Данный подход существенно сокращает время разработки БИУС за счет распараллеливания процессов разработки аппаратной части, программного обеспечения и диагностического обеспечения БИУС.

Ключевые слова: бортовая информационно-управляющая система, отладка программного обеспечения, виртуальный стенд, моделирование.

Nikolay Alexeevich Dolbnya, post-graduate student; graduated from the Faculty of Mechanics and Mathematics of Lomonosov Moscow State University; head of a project design team at PJSC 'Ulyanovsk Instrument Manufacturing Design Bureau'; author of publications in the field of computer-aided design of onboard information systems. e-mail: dolbnya_na@ukbp.ru.

Kirill Valentinovich Larin, Candidate of Engineering; graduated from the Faculty of Engineering and Physics of Leningrad Institute of Fine Mechanics and Optics; chief designer at PJSC 'Ulyanovsk Instrument Manufacturing Design Bureau'; author of publications in the field of computer-aided design of onboard information systems. e-mail: lucky@ukbp.ru.

Vadim Viktorinovich Shishkin, Candidate of Engineering; graduated from the Faculty of Radioengineering of Ulyanovsk Polytechnic Institute; Professor of the Chair 'Measuring and Computing Complexes' at Ulyanovsk State Technical University; author of articles in the field of the development of CAD, intellectual systems, real-time operating systems and embedded systems. e-mail: shvv@ulstu.ru.

Abstract

The article describes a method for the creation of virtual equipment for aircraft onboard information-management system (the System). The virtual equipment is implemented as a client-server application in a local area network. It enables the debugging of operational software for the System without onboard computer. In this case the physical layer of the system communication channels is replaced by the TCP transport layer. The method also enables the development and debugging of the System diagnostics. This approach reduces substantially time for the System development due to multisequencing of processes of the development of the System hardware, software and diagnostics.

Key words: onboard information-management system, software debugging, virtual equipment, modeling.

ВВЕДЕНИЕ

Современные авиационные бортовые информационно-управляющие системы имеют сложную гетерогенную структуру и в общем случае представляют собой сложноподчиненную совокупность блоков и индикаторов, объединенных линиями связи между собой, а также с датчиками и другими бортовыми системами, выступающими в качестве источников сигналов. Блоки и индикаторы выполняются на базе мощных процессорных модулей с разработанным функциональным программным обеспечением (ПО), работающих под управлением операционной системы жесткого реального времени (операционной системы целевой платформы).

В силу сложности ПО БИУС затраты на его разработку часто превышают затраты на разработку аппаратной части системы. Затраты и время на разработку ПО еще более увеличиваются в связи с необходимостью его сертификации по одному из принятых в гражданской авиации международных стандартов DO-178 [1].

Одним из путей повышения эффективности процесса создания БИУС является параллельная разработка аппаратной части и ПО. Такой подход позволяет существенно сократить время от момента формулирования требований заказчиком БИУС до момента ввода БИУС в эксплуатацию. Однако для отладки ПО в составе БИУС необходим полноценный полунатурный стенд, состоящий из стоек, вычислителей, электрических установок, специальных соединительных жгутов, а также станции контроля. Проектирование и создание этого стенда, а также решение вопросов коммутации составных частей стенда требуют существенного дополнительного времени.

Таким образом, для организации разработки ПО БИУС параллельно с ее аппаратной частью требуется решение проблемы отладки ПО на технологическом компьютере с учетом всех взаимодействий элементов в сложной гетерогенной структуре БИУС в условиях отсутствия полунатурного стенда.

СПОСОБ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

Для решения данной задачи предлагается использовать виртуальный стенд, учитывающий специфику всей используемой аппаратуры требуемого полунатурного стенда и моделирующий физическую среду, в которой реализованы каналы связи между отдельными составляющими БИУС, а также между ПО системы и диагностическим ПО.

Основой такого виртуального стенда является программная модель вычислителя (ПМВ), представляющая собой автономное приложение для настольной операционной системы технологического компьютера разработчика ПО, например, Windows. Это приложение имитирует поведение вычислителя и отдельных его контроллеров, по возможности повторяя все его временные характеристики, важные при разработке ПО, а также моделируя все его интерфейсы с ПО. Обмен данными по линиям между вычислителями БИУС (RS-232/422/485, LVDS, ARINC-429, ARINC-717, ARINC-708 [2-4]) в этом случае моделируется обменом данными по каналам, доступным для настольно-

го компьютера разработчика ПО (RS-232, Ethernet, механизмы межпроцессного взаимодействия). При этом настройке каналов обмена (скорость передачи, число бит данных в слове, контроль целостности данных и т.п.) инкапсулируются протоколом обмена между ПМВ. Созданную ПМВ можно протестировать при помощи контрольно-проверочной программы, разрабатываемой параллельно с ПМВ, начиная с момента утверждения интерфейсов, предоставляемых пакетом поддержки вычислителя (Board Support Package, BSP), а также пакетом поддержки процессорного модуля (BSP модуля).

При помощи ПМВ также моделируются интерфейсные платы программно-диагностического комплекса, для которого разрабатывается диагностическое ПО системы.

Основной особенностью ПМВ при этом является возможность использовать ее для моделирования аппаратуры в реальном режиме времени, т.е. все временные характеристики до уровня циклограмм реальной БИУС, а также полунатурного стенда будут сохранены и учтены при проектировании ПО БИУС.

ОСОБЕННОСТИ РАЗРАБОТКИ ПО БИУС С ИСПОЛЬЗОВАНИЕМ ПМВ

При разработке ПО БИУС в этом случае нужно учитывать ряд аспектов. Для реализации взаимодействия ПО с ПМВ необходимо выбирать любой из доступных механизмов межпроцессного взаимодействия технологической операционной системы: безымянные и именованные каналы, мейлслоты, сокет, разделяемые области памяти, сигналы и программные прерывания. Выбранный механизм должен не только удовлетворять требованиям к скорости взаимодействия, но также и быть более близким по реализации к механизму взаимодействия с реальным вычислителем.

Следующим важным аспектом является обеспечение переносимости ПО между технологической и целевой платформами на уровне исходного кода. Т.е. разработанное и отлаженное на технологическом компьютере ПО должно без изменений перекомпилироваться для целевой платформы. Это требование обеспечивается использованием системных вызовов только стандарта POSIX [5] и виртуализацией непосредственных обращений к устройствам вычислителя.

При отладке ПО индикатора ПМВ должна стыковаться с модельной версией графической библиотеки для вывода изображения на экран компьютера, поддерживающей те же интерфейсы, что и графическая библиотека для индикатора в составе БИУС.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПМВ

В общем случае ПО вычислителя БИУС, независимо от платформы реализации, имеет архитектуру, показанную на рисунке 1. Пунктирной стрелкой обозначены вызовы, которые присутствуют в случае с реальным вычислителем и отсутствуют в случае с ПМВ. Серым цветом выделено множество программных модулей, полученных из исходного кода, совпадающего с исходным кодом для целевого вычислителя. Стрелками обозначено направление вызова функций. Большая часть модулей BSP процессорного

модуля также не изменяется при использовании ПМВ. Изменению подвергается только модуль, реализующий функции доступа к аппаратуре вычислителя.

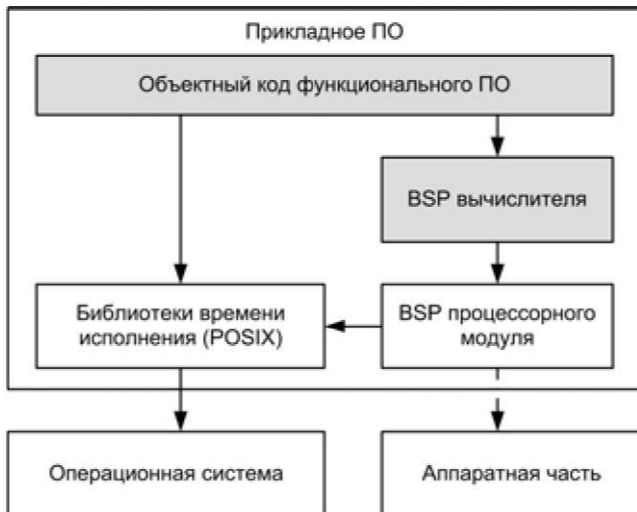


Рис. 1. Архитектура ПО вычислителя БИУС

Наиболее универсальным решением для программного модуля, моделирующего доступ к устройствам процессорного модуля, является использование стандартных TCP-соединений, поскольку все настольные рабочие станции имеют сетевые интерфейсы со скоростью передачи до 1 Гбит/с, что превосходит скорость передачи данных в любых каналах, используемых на борту самолета. Исходя из этих же соображений TCP-соединения могут использоваться для моделирования каналов обмена данными между вычислителями БИУС, а также между процессорным модулем вычислителя и его оконечными устройствами, такими как модуль лицевой панели индикатора, модуль пространственной ориентации и т.п. Более того, TCP-соединения могут устанавливаться не только между разными хостами локальной вычислительной сети, но и внутри одного хоста, оснащенного виртуальным сетевым интерфейсом логической петли localhost. Далее будем отождествлять доступ к устройствам процессорного модуля с доступом к слотам шины, на которой расположены эти устройства, реализуя логику взаимодействия «запрос с подтверждением». Вместо инициализации драйвера доступа к шине в модельном BSP процессорного модуля должно устанавливаться соединение с сервером ПМВ. Остальная логика BSP остается без изменений и отлаживается в тесном взаимодействии с ПМВ.

При таком подходе ПМВ будет выступать серверным приложением, моделирующим устройства, расположенные на шине. Так как в процессорных модулях вычислителей БИУС используются процессоры с модулем управления памятью (Memory Management Unit, MMU), позволяющим транслировать диапазон физических адресов процессора в обращения к внешним устройствам, то основным свойством процессорного модуля, используемым при реализации ПМВ, является то, что регистры и память устройств на шине представлены в виде диапазонов в адресном пространстве центрального процессора. Это позволяет сер-

веру ПМВ хранить копию адресного пространства шины в оперативной памяти. Эта область оперативной памяти может быть доступна из любой части сервера ПМВ по указателю, что позволяет реализовывать модели отдельных контроллеров устройств на шине в виде подключаемых динамически загружаемых модулей сервера ПМВ. Характеристиками каждого такого модуля являются: диапазон адресов на шине; указатель на точку входа init; указатель на точку входа write; указатель на точку входа read.

Диапазон адресов подключаемого модуля использует сервером ПМВ для определения, к какому подключаемому модулю транслировать операции чтения и записи. Важно, что, поскольку адреса регистровой зоны устройства и памяти устройства могут быть разнесены, регистры и память должны быть представлены отдельными подключаемыми модулями. Это оправдано также и тем, что доступ к регистрам устройства по функциональному назначению отличается от доступа к его памяти.

Указатель на точку входа init предоставляется подключаемым модулем для имитации полного сброса контроллера. Также эта функция вызывается при старте сервера. Точки входа read и write используются сервером ПМВ при запросе прикладной программой операций чтения и записи слов из устройства. Это связано с тем, что каждый регистр каждого устройства может реагировать на чтение и запись по-разному: некоторые регистры могут быть доступны только на чтение или только на запись, запись в некоторые регистры приводит к выполнению устройством сброса и т.п.

При разработке ПМВ все подключаемые модули должны находиться в отдельной папке, которая сканируется сервером ПМВ. Использование механизма подключаемых модулей позволяет моделировать самые различные устройства вычислителя БИУС, не изменяя сервера ПМВ. Таким образом, имитацией выдачи и приема данных и моделированием протоколов обмена в составе БИУС занимаются именно подключаемые модули под управлением сервера ПМВ.

При проектировании ПМВ особое внимание уделяется сохранению временных характеристик работы контроллеров модуля, таких как паузы между выдаваемыми словами, реальная скорость передачи по физическому каналу связи и т.п. Сохранение характеристик реализуется не в самом сервере ПМВ, а в подключаемых модулях, моделирующих отдельные устройства. Полная аутентичность виртуального стенда при этом гарантируется только при реализации каждого контроллера в отдельном программном потоке под управлением отдельного процессора. Эта проблема решена благодаря развернутости ПМВ в сетевой среде, так как связь между моделями отдельных контроллеров реализована так же, как и связь между несколькими ПМВ - в локальной вычислительной сети.

РЕАЛИЗАЦИЯ ПОДКЛЮЧАЕМОГО МОДУЛЯ ПМВ НА ПРИМЕРЕ КОНТРОЛЛЕРОВ ПРИЕМА И ВЫДАЧИ СЛОВ ПО СТАНДАРТУ ARINC-429

Структура подключаемых модулей определяется особенностями физических каналов связи, реализуемых контроллерами, которые моделируются данными под-

ключаемыми модулями. Например, для ARINC-429 данные передаются из одной выходной линии на множество входных, это означает, что настройками подключаемого модуля, моделирующего контроллер входных линий, будут только сетевой интерфейс и номер TCP-порта, на котором ожидается соединения для передачи данных. Этот подключаемый модуль, как и реальный контроллер, не будет сигнализировать об успешности приема, т.е. соединение будет однонаправленным.

Так как данные могут быть успешно приняты контроллером только тогда, когда передающая и принимающая линии настроены на одну скорость, значение скорости передачи должно быть инкапсулировано протоколом обмена. То же справедливо и для настроек проверки целостности данных дополнением до четности/нечетности. Кроме того, для упрощения и большей гибкости настроек принимающий модуль получает все данные на один порт, это означает, что протокол обмена должен поддерживать возможность демультимплексирования. Соответственно, структура пакета данных, принимаемых подключаемым модулем, моделирующим контроллер входных кодовых линий связи по ARINC-429, имеет формат, представленный в таблице 1.

Важной особенностью реализации подключаемого модуля контроллера выходных линий связи является использование асинхронных сокетов, так как при использовании синхронных сокетов в условиях недоступности принимающей стороны каждая операция передачи данных будет сопровождаться попыткой установки соединения. А при отсутствии возможности установить соединение эта попытка может занять несколько секунд, что при моделировании системы недопустимо, так как в реальном устройстве выдача производится независимо от наличия оконечного принимающего устройства.

МЕТОДИКА СОЗДАНИЯ И КОНФИГУРИРОВАНИЯ ВИРТУАЛЬНОГО СТЕНДА

При таком подходе виртуальный стенд БИУС является развернутым в локальной вычислительной сети приложением с использованием сконфигурированных серверов ПМВ. Под конфигурированием понимаются не только настройка сервера на используемый сетевой интерфейс и TCP-порт, но и настройки всех подключаемых модулей, моделирующих контроллеры бортового вычислителя. Пример структуры виртуального стенда приведен на рисунке 2.

Таблица 1

Формат пакета данных

Смещение в байтах относительно начала пакета	Размер поля пакета в байтах	Назначение поля пакета
0x00	0x01	Номер способа контроля целостности данных (0 - дополнение до нечетности, 1 - дополнение до четности)
0x01	0x01	Номер скорости передачи данных (0 - скорость 12.5 Кбит/с, 1 - скорость 50 Кбит/с, 2 - скорость 100 Кбит/с)
0x02	0x01	Номер входной линии, в которую передаются данные
0x03	0x04	Слово данных по стандарту ARINC-429

Для простоты отладки контроль целостности данных и скорость можно не учитывать, так как в реальных БИУС скорость определяется заранее еще на этапе проектирования системы в части определения межблочных интерфейсов, а в качестве контроля целостности почти всегда используется только дополнение до нечетности и многие контроллеры даже не позволяют менять этот способ.

Настройки подключаемого модуля выходных линий включают в себя для каждой выходной линии множество параметров вида:

XXX:PPP: NNN,

где *XXX* - IP-адрес модели контроллера входных линий;

PPP - номер порта этого контроллера;

NNN - номер входной линии на этом контроллере.

То есть информация о конфигурации линий связи системы содержится в настройках выходного контроллера. Это же справедливо и для контроллеров, реализующих выдачу информации по каналам других типов.

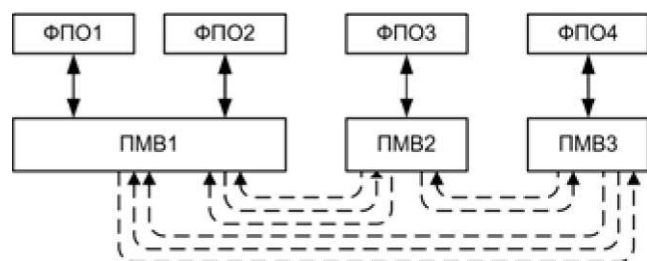


Рис. 2. Пример структуры виртуального стенда

Здесь ФПО - функциональное ПО, представленное моделью, скомпонованной с соответствующим модельным BSP. ФПО1 и ФПО2 моделируют два процесса, функционирующих на одном вычислителе. Моделируемая система состоит из трех вычислителей. Жирными двунаправленными стрелками обозначены соединения с ПМВ для доступа к устройствам вычислителя. Однонаправленные пунктирные стрелки обозначают обмен данными, моделирующий взаимодействие по каналам межблочного обмена. Каждая

Трудозатраты на разработку ПО БИУС по системам

Номер БИУС	Объем исходного кода ПО БИУС в строках	Затраты на разработку ПО БИУС в чел./ч	Объем комплекта диагностического ПО в строках	Затраты на разработку диагностического ПО в чел./ч	Выигрыш от использования виртуального стенда, чел./ч
1	≈40000	-500	-4000	-250	-200
2	≈75000	-630	-5500	-320	-350
3	-38000	-420	-2000	-210	-180
4	-53000	-480	-3000	-270	-270

стрелка должна конфигурироваться. В качестве конфигуратора можно использовать специальное приложение с графическим пользовательским интерфейсом, результаты работы которого удобно сохранять в базе данных настроек системы. Как правило, эта работа выполняется на этапе формирования требований к БИУС.

Важной особенностью виртуального стенда на базе ПМВ является возможность использования для отладки и тестирования ПО программно-диагностического комплекса (ПДК) «Фрегат» [6]. При этом разработанные тесты могут без изменений использоваться при диагностировании реальной системы в составе полунатурного стенда БИУС, что дает значительный выигрыш во времени разработки. Для этого разработан ряд драйверов виртуальных устройств «Фрегат», представляющих собой подключаемые модули менеджера устройств ПДК. Эти подключаемые модули выполнены в виде динамически загружаемых библиотек согласно спецификациям менеджера устройств. Драйвер виртуального устройства устроен так, что при первом обращении к нему запускается сервер ПМВ, сконфигурированный специальным образом: подключаемые модули сервера ПМВ выбраны таким образом, что данный экземпляр моделирует уже не бортовой вычислитель, а отдельную интерфейсную плату ПДК «Фрегат». Соответственно менеджер устройств ПДК «Фрегат» в данном случае является клиентом множества серверов ПМВ, а его подключаемые модули компонуется с модельным BSP, предоставляющим доступ к ПМВ.

Драйверы виртуальных устройств разработаны с учетом протокола обмена с ними менеджера устройств ПДК «Фрегат»: они поддерживают такие же имена оконечных устройств, но вместо обращений к интерфейсной плате содержат обращения к ПМВ. Масштабируемая и расширяемая за счет подключения дополнительных менеджеров устройств ПДК «Фрегат» в локальной вычислительной сети позволяет использовать такие драйверы виртуальных устройств без каких бы то ни было ограничений.

С учетом того, что характеристики реальных интерфейсных плат ПДК «Фрегат» могут отличаться от характеристик аналогичных контроллеров в ПМВ, подключаемые модули, моделирующие эти контроллеры, могут отличаться от реальных контроллеров регистровой моделью. Поэтому в составе виртуального стенда могут использоваться как связка BSP с подключаемыми модулями, обеспечивающая идентичность с моделируемым объектом на уровне регистровой модели, так и аналогичная связка, отличающаяся

от реального контроллера на уровне регистровой модели, но обеспечивающая идентичную функциональность.

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА ВИРТУАЛЬНОМ СТЕНДЕ

В рамках испытаний разработанной методики организации виртуального стенда были применены четыре разные БИУС, используемые в трех разных гражданских самолетах и вертолетах. ПМВ дорабатывалась в части подключаемых модулей только при создании виртуального стенда первой системы и далее уже не подвергалась модификации. Во всех системах были задействованы следующие физические каналы связи межмодульного и межблочного обмена: RS-422/485, дискретные сигналы, UART16550A, ARINC-429, ARINC-717 и ARINC-708A.

Группа разработчиков ПО БИУС состояла из трех человек, доработками ПМВ занимался один разработчик. Порядок трудозатрат представлен в таблице 2.

На разработку подключаемых модулей, реализующих модели приемников и передатчиков данных линий, было затрачено около 250 человеко-часов, на разработку драйверов ПДК «Фрегат» соответствующих виртуальных устройств - около 170 человеко-часов. Временные затраты на конфигурирование виртуального стенда столь малы, что могут быть исключены из рассмотрения.

В результате с использованием виртуальных стендов систем первая версия ПО даже для первой из проектируемых БИУС была получена на месяц раньше, чем для аналогичной системы с использованием только полунатурного стенда. На полунатурном стенде БИУС ПО проходило только комплексное тестирование, в ходе которого не было выявлено ни одной недоработки, связанной с временными характеристиками аппаратуры. Таким образом, комплексная отладка при использовании виртуального стенда на базе ПМВ и ПДК «Фрегат» была сведена к минимуму.

Диагностическое ПО для ПДК «Фрегат» также не дорабатывалось на полунатурном стенде БИУС. Были выполнены лишь некоторые доработки драйверов виртуальных устройств ПДК «Фрегат» по результатам тестирования диагностического ПО в рамках разработки первой из четырех БИУС.

ЗАКЛЮЧЕНИЕ

Таким образом, предложенный подход с учетом деталей его реализации успешно решает поставленные задачи сокращения временных затрат на разработку БИУС за счет распараллеливания процессов разработки аппаратной и

программной составляющих БИУС, а также процесса разработки диагностического ПО для использования в составе полунатурного стенда совместно с ПДК «Фрегат».

Моделирование каналов обмена между устройствами в сетевой среде гарантирует аутентичность результатов тестирования ПО при отсутствии необходимости выполнять дополнительные сетевые настройки и настраивать таблицы маршрутизации. Моделирование гетерогенной среды каналов обмена между вычислителями БИУС в гомогенной среде обеспечивает возможность реализации любого протокола обмена и отладки механизма его реализации в условиях отсутствия полунатурного стенда БИУС.

Затраты на разработку сервера ПМВ, его подключаемых модулей, модельных версий BSP вычислителя и BSP процессорного модуля, а также сложность конфигурирования виртуального стенда БИУС являются не столь значительными в условиях разработки множества БИУС на базе схожих процессорных модулей. Тем более, что конфигурирование каналов обмена виртуального стенда идентично конфигурированию системы на стенде, а конфигуратор виртуального стенда может использоваться также для конфигурирования реального стенда системы в автоматизированном режиме вместо конфигурирования его вручную.

СПИСОК ЛИТЕРАТУРЫ

1. Software Development under DO-178B. - Режим доступа: <http://www.opengroup.org/rtforum/jan2002/sLides/safety-critical/chiLenski.pdf>.
2. ARINC-429. - Режим доступа: <http://www.arinc.com/>.
3. ARINC-717. - Режим доступа: <http://www.arinc.com/>.
4. ARINC-708. - Режим доступа: <http://www.arinc.com/>.
5. Информационная технология - интерфейс мобильной операционной системы (POSIX). Часть 1: Интерфейс прикладных программ (API) [Язык программирования C]. ISBN 5-93838-001-4. М.: Издание НИИСИ РАН, 1998, 1999.
6. Н.А. Долбня, В.В. Шишкин, С.В. Черкашин. Универсальная система диагностирования бортового радиоэлектронного оборудования. - Известия Самарского научного центра Российской академии наук. - 2009. - т. 11, №3(2). - С. 392-397.