

Леонид Преображенский, Лидия Процкая

Системы эксплуатационного контроля и аварийной регистрации (ЭКАР) полётной информации занимают особое место среди бортового радиоэлектронного оборудования летательного аппарата (ЛА).

ный спасаемый) и техническое обслуживание своих блоков. Основной особенностью систем ЭКАР является наличие наземной части, которая обеспечивает считывание, обработку и воспроизведение зарегистрированной в полёте информации, а также управление и визуализацию результатов работы при техническом обслуживании бортовых блоков.

Современная система ЭКАР является многоблочной, многофункциональной и обрабатывает большие потоки мультимедийной информации в РРВ. Программно-аппаратная реализация систем ЭКАР имеет следующие особенности:

- система непосредственно не влияет на безопасность полёта и качество решения функциональных задач ЛА, поэтому при выборе программно-аппаратной реализации возможно отступление от жёстких авиационных стандартов;
- система контроля должна быть значительно проще и дешевле, чем контролируемые устройства;
- бортовая и наземная части требуют интенсивного двухстороннего взаимодействия;
- специальное программное обеспечение (СПО) бортовой части системы состоит из 10–15% функционального ПО и 85–90% драйверов, обеспечивающих информационный обмен между источниками информации и функциональным ПО;
- пользователю наземной части удобно работать с привычным интерфейсом персонального компьютера.

Проектирование и эксплуатация систем ЭКАР осуществляется в следующих условиях:

- позднее начало разработки (работы начинаются, когда созданы основные системы авионики ЛА) и раннее

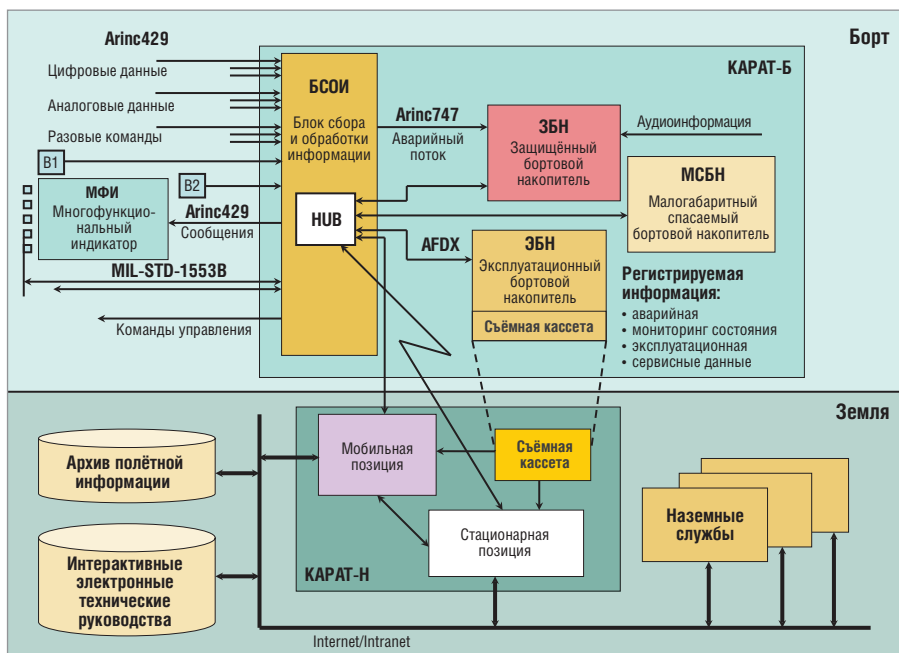


Рис. 1. Структурная схема современной системы ЭКАР

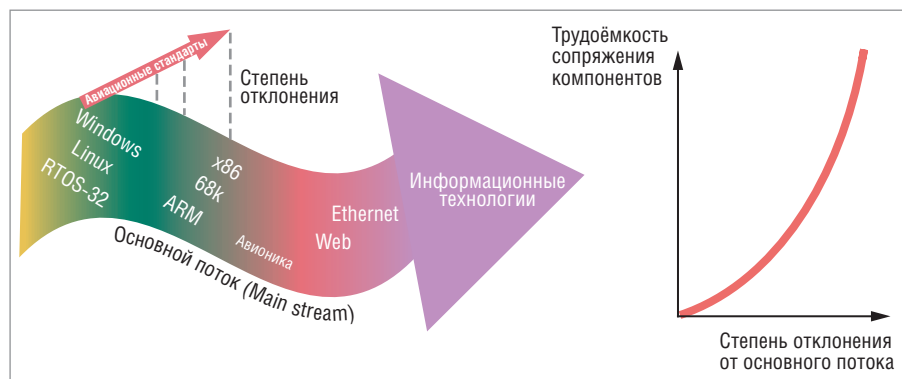


Рис. 2. Зависимость трудоёмкости сопряжения от степени отклонения компонентов от основного потока

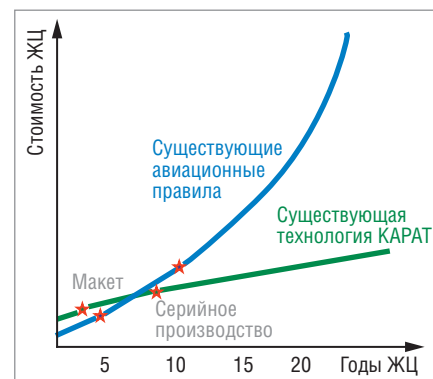


Рис. 3. Зависимость стоимости жизненного цикла от выбранной технологии

окончание проекта (ЛА должен быть укомплектован системой контроля и регистрации к первому полёту);

- разработка системы длится 5–7 лет, эксплуатация 20–25 лет. За это время существенно меняются доступные аппаратные и программные компоненты;
- малая серия и малое финансирование;
- несколько фирм-разработчиков с собственными технологиями проектирования;
- итерационный процесс проектирования — методы и алгоритмы контроля невозможно сформулировать в начале процесса разработки, они существенно меняются за время создания.

Принимая во внимание требования к программно-аппаратной реализации систем ЭКАР и условиям их проектирования, при определении облика систем КАРАТ была выбрана стратегия использования имеющихся на рынке коммерческих программно-аппаратных компонентов. Основными требованиями к этим компонентам являются следующие:

- нахождение в основном потоке (main stream);
- соответствие открытым международным стандартам;
- совместимость настольных и встраиваемых (embedded) технологий;
- интерфейс ОС с разрабатываемым ПО должен быть известным и хорошо документированным;
- возможность применения для разработки ПО инструментальных средств нескольких независимых разработчиков;
- для программных компонентов — наличие открытого кода.

ПРИНЦИПЫ ВЫБОРА ПРОГРАММНО-АППАРАТНОЙ РЕАЛИЗАЦИИ СИСТЕМ ЭКАР

Прежде чем приступить к анализу опыта применения конкретной ОС RTOS-32 в системах КАРАТ, авторы считают необходимым изложить принципы выбора программно-аппаратной реализации систем ЭКАР, которые были сформулированы в ходе дебатов с коллегами по этому вопросу.

15-летний опыт создания систем КАРАТ показывает, что стоимость жизненного цикла (ЖЦ) систем ЭКАР существенно зависит от выбранных при проектировании программно-аппаратных и технологических решений.

За время ЖЦ систем ЭКАР (25–32 года) согласно эмпирическому закону Мура число транзисторов на кристалле удваивается каждые 2 года и производительность цифровых систем экспоненциально возрастает. Вместе с электронными компонентами существенно меняются доступные на рынке информационные технологии. На протяжении ЖЦ системы приходится многократно осуществлять замену программно-аппаратных компонентов, выполнять повторную отладку и верификацию системы. Сложность и стоимость процесса повторной отладки напрямую зависят от архитектуры системы и совместимости применяемых при проектировании технологий с технологиями будущих реализаций. Чем более узкоспециализированными были технологии и правила

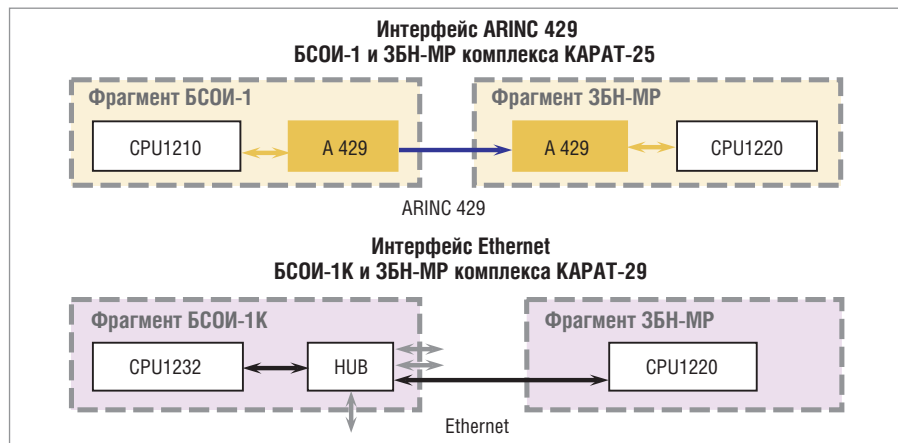


Рис. 4. Примеры реализации бортовых межблочных интерфейсов

взаимодействия компонентов системы в начале проектирования, тем выше вероятность, что они «разойдутся» с будущими технологиями и правилами, формируемыми рынком. Следствием этих отличий являются значительные усилия, которые необходимо приложить для сопряжения компонентов (рис. 2).

Для минимизации затрат на ЖЦ систем ЭКАР необходимо при проектировании следовать следующим принципам:

Первый принцип построения системы ЭКАР. Минимизация числа точек сопряжения и типов взаимодействия сопрягаемых компонентов в системе.

Второй принцип построения системы ЭКАР. Применение в точках сопряжения наиболее широко используемых открытых стандартов и технологий, доступных на рынке компонентов основного потока, то есть использование коммерческих аппаратных и программных средств, по возможности COTS-решений.

На рис. 3 приводятся графики стоимости ЖЦ систем ЭКАР ЛА для двух вариантов технологий — традиционной, базирующейся на узкоспециализированных авиационных правилах, и существующей технологии KARAT, базирующейся на COTS-решениях и открытых международных стандартах.

При проектировании системы по традиционной авиационной технологии начальные затраты ниже за счёт использования специализированных компонентов и авиационных правил. Это утверждение справедливо, если на момент начала проекта правила еще не успели устареть. Со временем необходимость в модификации системы приводит к экспоненциальному росту стоимости её ЖЦ, вызванному высокой трудоёмкостью сопряжения авиационных и неавиационных компонентов.

Стоимость модификации на протяжении ЖЦ системы, построенной на

основе технологии KARAT, базирующейся на COTS-решениях и открытых международных стандартах, значительно ниже за счёт менее трудоёмкой модернизации системы. Низкая трудоёмкость модернизации системы обусловлена следующими причинами:

- компоненты основного потока проходят тестирование миллионами пользователей, в то время как уникальные авиационные компоненты часто выдаются потребителю с ошибками, поскольку тестируются очень маленькими коллективами;
- новые компоненты основного потока выпускаются, как правило, с обеспечением совместимости со старыми компонентами основного потока;
- идеологически различные компоненты основного потока совместимы, и их сопряжение не требует больших трудозатрат.

Из первого принципа следует *правило минимизации числа адаптеров (аппаратных преобразований)*. На рис. 4 приведены для сравнения реализации бортовых межблочных интерфейсов — старого авиационного интерфейса ARINC 429 и современного стандартного Ethernet.

Общение двух современных процессоров по ARINC 429 по интерфейсу с 30-летней историей можно сравнить с переводом текста, написанного на одном из западноевропейских языков, на древнеегипетский язык и обратно. Так как в древнем языке нет ряда понятий современного языка, то такая конвертация данных неизбежно приводит к искажению семантики, а степень искажения зависит от разности языков и идеологий. И чем больше отличаются языки, тем дороже конвертация в процессе работы и более трудоёмка отладка взаимодействия при модернизации системы. Стоит также иметь в виду, что на сложные конвертации данных расходуются ресурсы процессора,

что предъявляет повышенные требования к характеристикам вычислителя.

В нашем примере Ethernet — это язык, понятный современным процессорам и не требующий перевода на аппаратном уровне COTS-модуля.

Таким образом, при проектировании систем необходимо выбирать те языки (технологии), которые требуют минимальных преобразований на аппаратном уровне.

Аналогичное правило формулируется и для программного обеспечения — *минимизация числа преобразований на программном уровне*. В настоящее время для обеспечения адаптации ПО к смене программно-аппаратных компонентов в течение ЖЦ авиационных систем разрабатывается технология интегрированной модульной авионики (ИМА), схематически представленная на рис. 5. Суть этой технологии в отделении приложений от среды выполнения (аппаратура, ОС, драйверы) и обеспечении общения приложений со средой выполнения по POSIX-стандарту. Тогда при переходе к новой реализации необходимо настроить интерфейсы между приложениями и средой выполнения. Такая технология не оправдана для систем ЭКАР, у которых 85–90% СПО составляют интерфейсы, а алгоритмическое обеспечение (приложение) составляет всего 10–15%. В этом случае для исполнения сравнительно небольшого приложения задействованы большая операционная система и сложная система драйверов, требующая много ресурсов.

Для систем ЭКАР следует проводить программную адаптацию к смене компонентов на уровне системы (приложение + ОС + драйверы как единое целое, генерируемое в ОС OnTime RTOS-32), схематически технология программной адаптации показана на рис. 6. Такой подход позволит значительно снизить затраты на отладку системы в целом, он существенно менее требователен к ресурсам процессора в момент исполнения и не противоречит принятой процедуре тестирования и верификации системы при смене поколений аппаратуры.

Следующим правилом, вытекающим из первого принципа и позволяющим минимизировать стоимость ЖЦ наземно-бортовой системы ЭКАР, является применение *технологии совместной разработки бортовой и наземной частей системы*. Основой технологии стало использование совместимых платформ (процессор + ОС) бортовой и наземной частей комплекса и совместная разра-

ботка, отладка и сопровождение бортового и наземного программного обеспечения. Технология предполагает:

- совместную разработку бортовой и наземной программных частей с общими программными модулями контроля и методами их верификации;
- сеть с протоколами реального времени для межблочного бортового обмена и обмена с наземной частью;
- инкапсуляцию обрабатываемых данных и их описаний, помещение всех необходимых для дальнейшей корректной обработки данных в бортовую часть;
- обслуживание бортовой части только с помощью наземной части без дополнительной аппаратуры;
- механизм эволюционного перемещения функций контроля наземной части в бортовую часть по мере развития систем ЭКАР;
- автоматизацию проектирования на основе электронного документооборота между разработчиком алгоритмов контроля и ПО контроля.

Применение перечисленных принципов позволило создать наземно-бортовую систему контроля и регистрации полётной информации нового поколения КАРАТ, реализующую:

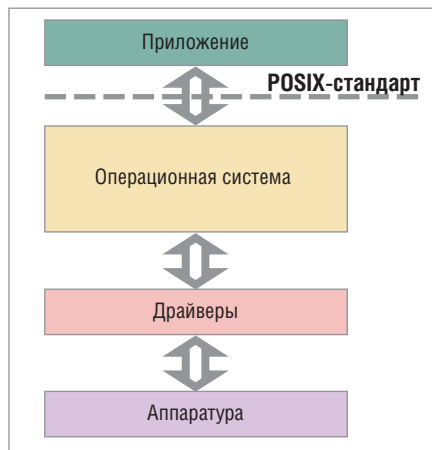


Рис. 5. Технология интегрированной модульной авионики

- интегрированную систему аварийной регистрации и эксплуатационного контроля для МиГ-21BIS UPG, Су-25СМ/УБМ, МиГ-29К в сжатые сроки при ограниченном финансировании;
- синхронную мультимедийную регистрацию полётной информации (параметрических, аудио- и видеоданных) на нескольких регистраторах для МиГ-29К;
- использование высокоскоростного канала обмена Ethernet;

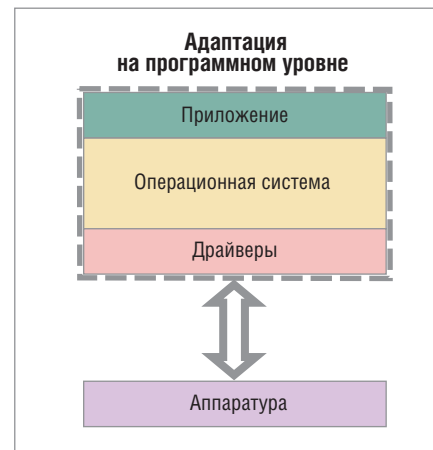


Рис. 6. Технология программной адаптации к смене компонентов на уровне системы

- новые возможности обслуживания бортовых блоков, а именно:
 - проведение технологических и градуировочных работ на самолёте;
 - мониторинг процессов сбора и передачи данных в режиме реального времени;
 - углублённый тест-контроль блоков КАРАТ в составе самолёта;
 - обслуживание памяти бортовых блоков КАРАТ;
 - замена версии ПО бортовых блоков КАРАТ;

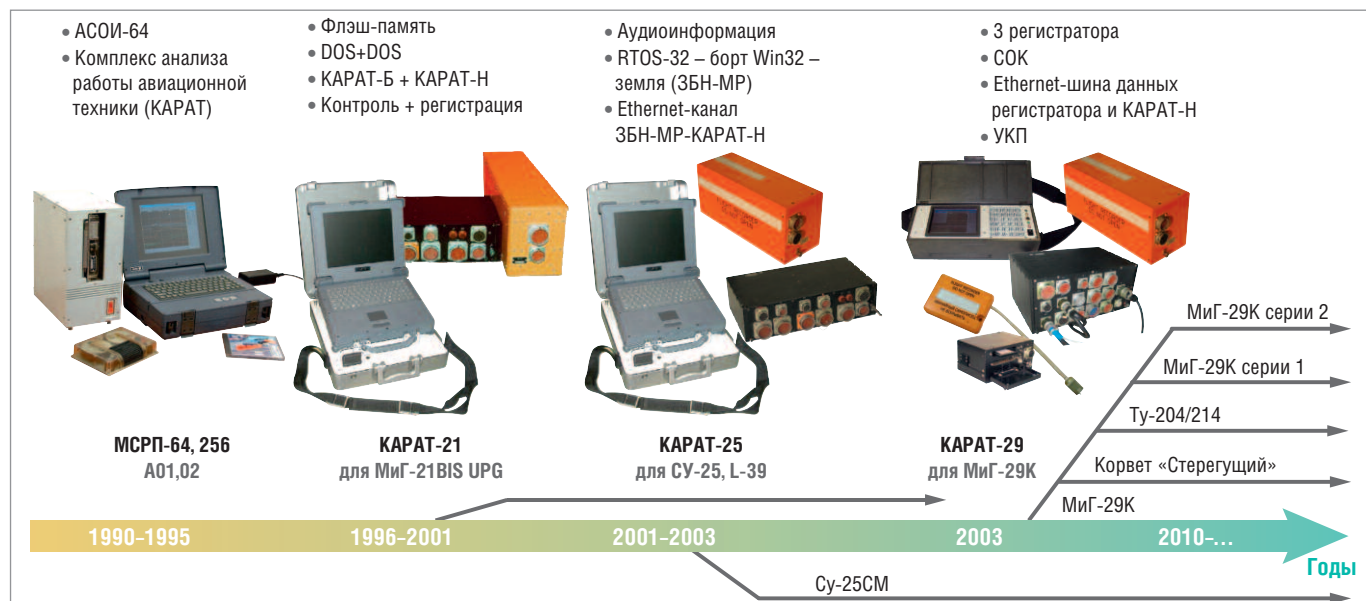


Рис. 7. История развития систем КАРАТ

— реализация симуляционного режима в серийном ПО, управляемого с наземной части.

Анализ опыта применения ОС On Time RTOS-32 в системах КАРАТ

На рис. 7 представлена история развития систем КАРАТ. В 90-х годах была создана наземная система АСОИ-64 для

обработки информации с кассетного бортового накопителя самолётов Ил-62М. Система была реализована на персональном IBM PC/AT совместимом компьютере с ОС Microsoft DOS.

Опыт создания наземных систем обработки полётной информации позволил перейти к разработке наземно-бортового комплекса контроля и регистрации полётной информации. Работы начались в феврале 1996 года. К концу года был готов макетный образец комплекса, а в ав-

густе 1997 года МиГ-21-93, оснащённый КАРАТ-21, совершил первый полёт. Успех проекта — создание в сжатые сроки при ограниченном финансировании комплекса с расширенными функциональными возможностями — во многом определялся выбором архитектуры бортовых блоков комплекса. Вычислители бортовых блоков были реализованы на процессорах типа PC/104 с ОС DOS, а наземная часть комплекса на IBM PC/AT совместимом компьютере с ОС Microsoft

Таблица 1

Сравнительные характеристики наземно-бортовых комплексов КАРАТ

№	Требования к программно-аппаратной реализации комплекса КАРАТ	РС (борт – земля)	DSP (борт) + РС (земля)	Примечания
1.	Выполнение требований существующих спецификаций и стандартов в области регистрации и обработки полётной информации	Да	Да	–
2.	Наличие проверенной технологии разработки, испытания и сопровождения (технологический задел)	Да	Нет	Опыт разработки АСОИ-64, КАРАТ-21 для МиГ-21 BIS UPG
3.	Обратная совместимость с КАРАТ для МиГ-21 BIS UPG	Да	Нет	–
4.	Интеграция бортовой и наземной частей в единый комплекс регистрации диагностики и обработки	Да	Нет	Единая платформа позволит сделать следующий шаг вперёд по сравнению с МиГ-21 BIS UPG, а разные платформы – это минимум один шаг назад
5.	Открытая система	Да	Нет	Открытая система состоит из компонентов, взаимодействующих друг с другом через стандартные интерфейсы
6.	Степень тестирования инструментальных средств разработки и технологий на наличие ошибок	Очень высокая	Низкая	Миллионы пользователей ежедневно тестируют РС-технологию
7.	Повторные вложения при моральном старении используемых информационных технологий в процессе выполнения проекта	Низкие	Очень высокие	Появление USB, Bluetooth и иных совместимых с РС бортовых технологий не потребует для единой платформы значительных вложений

DOS. ОС бортовых блоков была доработана для обеспечения режима реального времени. Такой выбор позволил проводить разработку и отладку ПО бортовых блоков на настольных компьютерах, максимально использовать опыт работы специалистов и обеспечить полный контроль параметрической информации бортовыми и наземными процессорами от момента приёма до анализа результатов. Работа над проектом продолжалась 5 лет, а после сдачи самолёта заказчику разработчики осуществляют сопровождение своей системы.

За пять лет работы над проектом большим коллективом специалистов была создана и реализована уникальная технология совместной разработки, отладки и сопровождения бортового и наземного ПО. Эта технология основана на опыте объединённой команды разработчиков из ФГУП «ГосНИИАС» (Москва), ОЗ «Прибор» (Санкт-Петербург), ОКБ «Авиаавтоматика» (Курск), МАПО МиГ (Москва), НАЗ «Сокол» (Нижний Новгород).

Ключевым условием для реализации технологии совместной разработки, отладки и сопровождения бортового и наземного ПО является правильный выбор ОС для бортовых и наземных компьютеров.

Для нового проекта (КАРАТ-25 для самолёта Су-25СМ/УБМ) было принято решение создать защищённый бортовой накопитель с расширенной памятью

(256 Мбайт против 10 Мбайт), и встал вопрос о замене ОС. Замена ОС DOS была вызвана следующими причинами:

- DOS перестала сопровождаться;
- DOS не поддерживала высокоскоростных протоколов обмена. Для считывания данных с бортового накопителя с расширенной памятью был необходим высокоскоростной канал считывания;
- в настольных компьютерах DOS заменили на Windows, которая стала наиболее широко применяемой ОС.

Если в первом проекте у нас не было времени для дискуссий по поводу выбора программно-аппаратной реализации комплекса КАРАТ и мы действовали интуитивно, то в новом проекте этому было уделено много внимания. Обсуждались два варианта аппаратной реализации. Это

- универсальная РС-технология для борта и земли;
 - специализированные DSP-процессоры для бортовых блоков и РС-технология для наземной системы.
- В таблицах 1 и 2 приведены сравнительные характеристики этих вариантов.
- Была выбрана универсальная РС-технология. Следующим шагом было определение конкретной ОС для применения в новом защищённом бортовом накопителе. Общие требования к ОС были следующие:
- КАРАТ-Б и КАРАТ-Н нового поколения должны соответствовать между-

Сравнительные характеристики бортовых частей КАРАТ

Таблица 2

№	Требования к программно-аппаратной реализации комплекса КАРАТ	PC (борт)	DSP (борт)	Примечание
1.	Потребляемая мощность и соответствующее тепловыделение	Больше	Меньше	В CPU1220 возможно управлять частотой, энергетикой и тепловыделением
2.	Цена процессорного модуля	Больше	Меньше	—
3.	Габаритные размеры (конструктив)	Больше и хуже	Меньше и лучше	—
4.	Время настройки (поиска неисправностей)	Меньше	Больше	—
5.	Возможность воспроизведения в настоящее время в России	Да	Да	Только из импортных покупных компонентов

народным и российским стандартам по системам аварийной регистрации и обработки полётной информации, а также требованиям потенциальных заказчиков;

- должна быть сохранена технология совместной разработки, отладки и сопровождения бортового и наземного ПО;
- желательно сохранить часть ПО на уровне текстов программ;
- интерфейс ОС с разрабатываемым ПО должен быть известным и хорошо документированным;
- должна существовать возможность применения для разработки ПО инструментальных средств нескольких независимых разработчиков;
- нужны исходные тексты, необходимые для сертификации разрабатываемого

ПО и для расширения функциональных возможностей ОС;

- требуется возможность установки ОС в существующие системы КАРАТ-Б и КАРАТ-Н.

В наземной системе предполагалось использовать широко распространённую в то время ОС Microsoft Windows 98, которая в то время устанавливалась на компьютер поверх DOS. Доработка функционального ПО в этом случае была незначительной.

К характеристикам ОС бортовой системы предъявлялись следующие требования:

- поддержка архитектуры процессора x86;
- интерфейс прикладного программирования — Win32;

- поддержка работы в жёстком реальном времени;
- запуск Windows-приложения без Windows;
- быстрый запуск ОС;
- ограниченный размер ПЗУ;
- многозадачность в рамках одного приложения.

Первоначально рассматривалось несколько десятков ОС. Для детального тестирования были выбраны три 32-рядные ОС с известным и хорошо документированным интерфейсом с прикладным ПО: Microsoft Windows NT Embedded 4.0, Microsoft Windows CE 2.0 и On Time RTOS-32. Сравнительные характеристики этих ОС приведены в таблице 3.

По результатам тестирования была выбрана ОС On Time RTOS-32. В 2000 году мы купили лицензию и начали работать на RTOS-32 версии 3.06. Отдельно было приобретено ПО фирмы EBS, обеспечивающее обмен информацией по протоколу TCP/IP и совместное с выбранной ОС.

Использование ОС On Time RTOS-32 позволило:

- сохранить технологию совместного проектирования и отладки наземной и бортовой частей комплекса, в частности, применять один и тот же компилятор для бортовых и наземных приложений;
- реализовать высокоскоростной канал Ethernet для считывания данных с защищённого бортового накопителя, который позволил увеличить скорость считывания по сравнению с RS-32 в десятки раз.

В следующем проекте — в четырёхблочной системе КАРАТ-29К для самолётов МиГ-29К индийских ВМС — ОС On Time RTOS-32 была загружена во все блоки системы. Для межблочного обмена в системе была реализована бортовая сеть Ethernet, к которой подключается компьютер наземной системы.

К настоящему времени коллектив из 11 человек разработал и сопровождает ПО 6 проектов. Кроме перечисленных, это ещё четырёхблочный КАРАТ-29К серии 2 для сирийских самолётов МиГ-29К, четырёхблочный КАРАТ-29К серии 1 для российских самолётов МиГ-29К, защищённый бортовой накопитель для корвета «Стережущий», защищённый бортовой накопитель для самолётов Ту-204/214. Во всех этих проектах для бортовых блоков использовалась ОС On Time RTOS-32. При сопровождении проекта периодически решаются следующие задачи:

Сравнительные характеристики операционных систем

Таблица 3

№	Требования к программно-аппаратной реализации комплекса КАРАТ	Microsoft Windows NT Embedded 4.0	Microsoft Windows CE 2.0	On Time RTOS-32
1.	Реальные требования к аппаратуре (процессор/ОЗУ/ПЗУ)	386/12 Мбайт/16 Мбайт	486/1 Мбайт / 1 Мбайт	386/1 Мбайт / 1 Мбайт
2.	Загрузка без BIOS	Нет	Да	Да
3.	ОС ориентирована на архитектуру	PC/AT	Не PC/AT (PC только как прототип CE-устройства)	PC/AT
4.	Среда разработки ПО	Microsoft Windows NT	Microsoft Windows NT	Microsoft Windows 95/98/NT
5.	Инструментальные средства разработки ПО	Microsoft, Borland, Watcom и др.	Только Microsoft	Microsoft, Borland, Watcom и др.
6.	Время загрузки на 486 DX2x66 ОЗУ 32 Мбайт	60–70 с	Секунды или доли секунды при загрузке без BIOS	Секунды или доли секунды при загрузке без BIOS
7.	Исходные тексты ОС	Нет	Частично	Да
8.	Сохранение разработанной технологии совместной разработки, отладки и сопровождения бортового и наземного ПО	Да	Нет	Да
9.	Сохранение текстов разработанного ранее ПО	20%	0%	60%
10.	Первоначальные затраты	\$10 000	\$10 000	\$20 000
11.	Затраты на каждую копию ОС при серийности 1000 и 10 экз.	\$100–\$1000	\$100–\$1000	Отсутствуют (оплачивается только лицензия разработчика)
12.	Наличие встроенного графического интерфейса пользователя	Да	Да	Нет

- адаптации ПО (бортового и наземного) к смене аппаратной реализации бортовых блоков новой серии, вызванной сменой поколений компонентов на рынке,
- перехода на новую ОС для наземной системы, вызванного сменой аппаратной реализации наземных компьютеров.

За 14 лет работы с ОС On Time RTOS-32 мы использовали версии 3.06, 4.19, 5.14, 5.21 и 5.26.

Технология разработки систем ЭКАР предполагает комплексную адаптацию приложения, драйверов и ОС к смене программно-аппаратной реализации системы. В случае необходимости мы корректируем код ОС. Благодаря наличию хороших заготовок в коде ОС On Time RTOS-32 нами были реализованы следующие функции:

- замена версии ПО бортовых блоков на основе расширения функций rtb-файла;
- динамическая загрузка исследовательского ПО в бортовые блоки;
- межблочный обмен по Ethernet-протоколу, реализованному поверх UDP (User Datagram Protocol – протокол пользовательских датаграмм);
- удалённый доступ к бортовым блокам ЛА, стоящего на земле.

Технология замены версий ПО бортовых блоков с наземной системы КАРАТ-Н реализована следующим образом:

- разбираем системный rtb-файл;
- подключаем загрузчик для новой версии ПО;
- создаём необходимую структуру размещения информации;
- отправляем в бортовой блок.

Навык ручной корректировки rtb-файлов и освоение технологии подмены rtb-файлов позволили нам реализовать динамическую загрузку исследовательского ПО в бортовые блоки.

Динамическая загрузка исследовательского ПО в бортовые блоки

Для реализации динамической загрузки ПО в бортовые блоки решены следующие задачи:

- обеспечение одновременного нахождения в памяти старого и нового процессов в ОС RTOS-32. В стандартных способах загрузки данных ОС RTOS-32 не предусмотрена возможность передачи управления от одного процесса другому. Для решения этой задачи обмен данными был реализован на основе виртуального диска, позволяющего иници-

ровать одновременно несколько процессов;

- реализация способа общения с виртуальным диском при минимизации времени загрузки данных. Для этого был создан протокол обмена данными с учётом минимизации времени загрузки данных;
- определение криптозащиты данных, которыми обмениваются процессы виртуальной машины и окружающих её реальных процессов. После проведения анализа способов криптозащиты данных был выбран способ, при котором каждый байт данных защищён паролем;
- оценка допустимости загрузки ПО на другие процессоры. В связи с многоблочностью системы ЭКАР предусмотрена проверка модулей памяти, позволяющая определить тип процессорного модуля и совместимость загружаемого ПО с данной моделью памяти;
- интеграция ПО динамической загрузки с существующей программой замены версии, входящей в состав штатного программного обеспечения.

Для реализации динамической загрузки создаётся ПО виртуального об- лака (виртуальной машины, перенесён-

ной по сети Ethernet). В бортовом блоке реализуется обмен данными старого и нового процесса.

Технология обмена данными между старым и новым процессами базируется на технологии создания виртуального диска, которая позволяет переносить все данные от старого процесса к новому. Виртуальный диск — это область памяти ОЗУ, создаваемая программными средствами для перенесения на него ПО и данных, в конкретном случае — нового процесса. Виртуальный диск создаётся в бортовом блоке.

Запуск нового ПО состоит из двух основных этапов:

- этапа инициализации;
- этапа поблочной загрузки.

Режим инициализации определяет создание виртуального диска в ОЗУ бортового блока и перенос на него данных для запуска нового процесса. Обмен производится по протоколу UDP. На виртуальный диск передаётся размер ПО и запрос о возможности его принятия. Если ответ на запрос положительный, то запускается следующий режим, если отрицательный, то пользователю выводится код ошибки.

Режим поблочной загрузки — это режим, при котором загрузка производит-

ся по 1 кбайт данных. В последнем килобайте данных находится служебная информация, которая включает, в том числе, контрольную сумму, флаги криптозащиты, проверку на наличие пользовательской лицензии.

Для проверки целостности программы используется 16-разрядный алгоритм расчёта контрольной суммы CRC-16 (cyclic redundancy code). Алгоритм CRC базируется на свойствах деления с остатком двоичных многочленов. Если проверка прошла успешно, то программа переходит к проверке криптозащиты. В случае несовпадения контрольной суммы пользователю выводится код ошибки.

Криптозащита в программе производится по паролю (каждый байт данных защищён паролем). Если проверка криптозащиты прошла успешно, то программа переходит к проверке прав пользователя. Если проверка прав пользователя не прошла успешно, то выдаётся код ошибки.

Если все этапы проверки прошли успешно, то управление передаётся новому процессу, который использует данные старого. При передаче управления должны быть определены конфигурационные данные системы КАРАТ (гра-

дуировка, описатель параметров и системная информация). Конфигурационные данные могут быть наследованы из старого процесса или перекрыты конфигурационными данными нового.

Загруженное таким образом исследовательское ПО работает до выключения или перезагрузки бортовой системы регистрации и контроля. После выключения/перезагрузки все динамически загруженные данные стираются и виртуальный диск исчезает из ОЗУ.

ЗАКЛЮЧЕНИЕ

Четырнадцатилетний опыт использования ОС On Time RTOS-32 подтвердил правильность выбора разработчиков. Всё это время ОС On Time RTOS-32 динамично развивалась в русле основных тенденций информационных технологий. Применение ОС On Time RTOS-32 позволяет обеспечивать длительный ЖЦ наземно-бортовых систем ЭКАР.

Предоставляемый код даёт возможность модернизировать реализованные в ОС функциональные задачи с учётом требований разрабатываемых систем.

Разработчики КАРАТ собираются создавать новые проекты с использованием ОС On Time RTOS-32. ●

leopreo@gmail.com