УДК 004.41

ПОДХОД К АДАПТАЦИИ СООБЩЕНИЙ ДЛЯ ОБМЕНА МЕЖДУ АВИАЦИОННЫМИ ПРИЛОЖЕНИЯМИ

[1, 2]Солоделов Ю.А., [1, 2]Камалетдинова Г.Р., [1]Кожанов К.Д., [1]Альбицкий Д.В., [1]Трубчанинов И.А.

[1]ФАУ «Государственный научно-исследовательский институт авиационных систем», Москва, e-mail: grkamaletdinova@2100.gosniias.ru;
[2]ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», Москва

Бортовые комплексы управления самолетов состоят из большого числа аппаратно-программных компонентов, взаимодействующих между собой, причем все большее число функций реализуется не отдельными вычислительными системами, а отдельными вычислительными процессами в одном вычислителе под управлением операционной системы. Взаимодействие вычислительных процессов осуществляется через обмен сообщениями. При этом возникает проблема согласованности форматов этих сообщений. Особенно актуальной является эта проблема в случае, когда функция, используемая повторно, уже была квалифицирована. В этом случае ее доработка с целью формирования необходимого вида сообщений потребует повторной квалификации, а значит, дополнительных затрат. Такой подход нецелесообразен. В статье рассматривается альтернативный подход к решению задачи унификации обмена между авиационными приложениями, который позволит упростить интеграцию повторно используемых функций, разработанных для использования операционной системой, поддерживающей стандарт ARINC 653 (Avionics Application Software Standard Interface). Для этого рассматриваются решения, предлагаемые в стандарте FACE (Future Airborne Capability Environment), и анализируются исходные данные. В работе предлагается использование специального программного обеспечения, которое будет выступать в качестве Адаптера обмена и обеспечит информационный обмен между пользовательскими приложениями. В статье описаны предпосылки перехода к такому решению и этапы развития проекта.

Ключевые слова: моделирование, унифицированный обмен, сервисное ПО информационного обмена (Адаптер), операционная система реального времени, бортовые системы, ОСРВ JetOS

APPROACH TO UNIFIED MESSAGING IN AVIATION APPLICATIONS

[1, 2]Solodelov Yu.A., [1, 2]Kamaletdinova G.R., [1]Kozhanov K.D., [1]Albitskiy D.V., [1]Trubchaninov I.A.

[1]GosNIIAS, Moscow, e-mail: grkamaletdinova@2100.gosniias.ru;
[2]Moscow Aviation Institute, Moscow

On-board aircraft control systems consist of a large number interacting hardware and software components. At the moment, the number of functions implemented as separated computing processes in the frame of one onboard computer controlled by the operating system is growing, comparing to the number of functions implemented on separated computing systems. Computing processes are intercommunicating by messaging. Hence the conformity of messages formats plays a crucial role. This problem becomes especially relevant in terms of functions qualification. E.g., if the function being reused has been already qualified, it cannot be easily modified in order to accept or transmit messages in another format due to complexity of the re-qualification procedures. Thereby this approach is inappropriate. The article discusses an alternative solution for unified messaging in aviation applications, which will simplify the integration of reusable functions designed for ARINC 653 standard (Avionics Application Software Standard Interface) operating systems. Discussed solutions are based on FACE (Future Airborne Capability Environment) standard and initial data analysis. The paper proposes to use special software that acts as an exchange adapter and implements unified messaging between functions. The article describes the idea of this approach and the project development.

Keywords: modeling, unified messaging, service software for information exchange (Adapter), real-time operating system, onboard systems, RTOS JetOS

Бортовой комплекс управления самолетом должен отвечать высоким требованиям по безопасности. Это достигается соблюдением нормативных требований и внешним контролем сертифицирующих органов [1]. При этом весь процесс разработки программного обеспечения строго регламентирован и отступление от него влияет на процесс сертификации, а значит, на возможность дальнейшего использования.

Выполнение целевых функций в комплексах бортового оборудования воздушных судов обеспечивается строгим соблюдением протоколов информационного взаимодействия между отдельными системами и/или функциональными программными приложениями. В связи с этим при появлении новых систем и/или функций возникает проблема обновления протоколов информационного взаимодействия [2] и их согласованности.

Разработка и квалификация новых функциональных программных приложений и систем является сложной, затратной

процедурой, поэтому усилия направляются на реализацию возможности повторного использования уже квалифицированных приложений: новый функционал может достигаться, к примеру, за счет их комбинации. В этом случае протоколы информационного взаимодействия не будут совпадать с исходными.

Отсюда возникает проблема не просто обновления протоколов (что потребует изменения приложений), а их унификации [3, 4], чтобы формат сообщений мог быть адаптирован под любые взаимодействующие приложения.

При условии неизменности приложений требуется разработка дополнительного программного средства, которое будет трансформировать сообщения необходимым образом.

В связи с этим была изучена практика подхода консорциума FACE [5], разрабатывающего стандарт, который направлен на упрощение интеграции повторно используемого программного обеспечения, как в новые проекты, так и в существующие рабочие проекты [6, с. 1–12].

В рамках данной работы рассматривался бортовой комплекс интегрированной модульной авионики и взаимодействие между приложениями [7–9].

Для моделирования процессов использовались архитектурные решения, предложенные FACE [6, с. 4–12], в рамках которых предлагается выделить сегмент транспортных служб, отвечающий за передачу сообщений [3; 6, с. 49–81]. Основываясь на этих данных и учитывая выбор стандарта ARINC 653 [10, с. 49–75], были сформулированы требования на разработку программного обеспечения, которое позволит реализовать унификацию протоколов информационного взаимодействия [11].

Программное обеспечение было реализовано, а затем включено в комплекс-демонстратор, позволяющий оценить работу программного обеспечения на моделируемых входных и выходных данных и проводить тестирование информационного взаимодействия функциональных программных приложений.

Данная работа направлена на поиск решения по адаптации обмена сообщениями между функциональными приложениями в условиях сертификационных ограничений на модификацию уже квалифицированных программных модулей. В условиях узкоспециализированной среды поиск решения ограничен, однако, анализируя опыт консорциума [5], а также опыт смежных специалистов [3; 4; 12], решение по унификации формата сообщений посредством специального программного обеспечения представляется наиболее логичным.

Следовательно, целью данной работы стала разработка программного обеспечения, которое служило бы для устранения имеющихся различий в форматах передаваемых/принимаемых сообщений. Для достижения данной цели разработано специальное программное обеспечение, которое включено в комплекс-демонстратор [13].

Материалы и методы решения

В программировании наиболее широко используемым способом организации данных является структура, которая включает в себя логически связанные наборы данных, в том числе и другие структуры. Для обмена информацией между приложениями чаще всего используются сообщения, представляющие собой структуры данных, где смысл данных обычно определяется местом в структуре [9, 12].

Порты передачи данных определяются ARINC 653 [10, с. 4–12], который был выбран для моделирования за удобство формирования сообщений.

Стандарт FACE требует, чтобы все структуры, которыми обмениваются приложения, строго соответствовали определенной модели данных [6, с. 82–86]. Реализация такого подхода требует либо создания приложений с учетом стандарта [6, с. 57–81], либо внесения изменений в уже созданные приложения.

В рамках данной работы требуется увязка уже существующего и квалифицированного функционального программного обеспечения, что накладывает существенные ограничения на возможность изменения этих модулей [1; 2].

Таким образом, необходимо обеспечить преобразование содержимого передаваемых сообщений так, чтобы оно соответствовало ожиданиям принимающего приложения без изменения самого приложения.

Данная задача решается, благодаря разработке специального программного обеспечения – Адаптера [3; 4; 12], который циклически получает и выдает информацию через порты ARINC 653 [10, с. 4–12], используя сервисы операционной системы реального времени [13] и обеспечивает трансформацию сообщений, не затрагивая структуру самого функционального программного обеспечения.

Таким образом, основным методом, используемым для решения поставленной задачи, является преобразование [3; 6, с. 49–73].

Возможными вариантами преобразований являются:

1. Преобразование размещения данных. Данный тип преобразования является наиболее очевидным при обмене структурами. Для реализации этого решения необходимо располагать информацией о порядке расположения данных внутри сообщений. Возможен вариант, когда необходимые выходные данные могут быть получены с помощью нескольких входных или быть частью сообщения, т.е. нет прямого соответствия между выходными и входными сообщениями и/или их частями.

2. Преобразование порядка выдачи данных.

3. Преобразования формата представления данных у источника и приемника. Различия могут быть существенными. Например, часто различается представление вещественных чисел – как по занимаемому размеру, так и по внутренней структуре представления мантиссы и порядка. Стандарт FACE полагает, что все вещественные числа должны быть представлены в формате IEEE 754 [14, с. 6–14] одинарной точности (32 разряда), двойной точности (64 разряда) или расширенной точности (80 разрядов). Может различаться и представление целых чисел в части числа занимаемых разрядов.

4. Преобразование единиц измерения в случае различных представлений данных на выход и прием.

5. Линейное преобразование чисел.

6. Преобразование начала и направления отсчета.

7. Преобразование кодировки (например, в случае обмена текстовыми данными).

8. Преобразование пересчета. Например, пересчет электрических показаний датчика в физические значения.

9. Преобразование систем отсчета (для систем координат или географических данных).

Все необходимые преобразования для настройки корректного информационного взаимодействия Адаптер будет исполнять, используя конфигурационные данные, сформированные заранее.

Адаптер должен допускать изменение значительного объема своих рабочих параметров – в первую очередь размера и числа портов и их характеристик, периодичность процессов, предназначенных для преобразования сигналов, а также состава структур, типов данных переменных в указанных структурах и алгоритмов преобразования [3; 6, с. 49–73]. Изменение рабочих параметров должно осуществляться без изменения исходного кода, с помощью конфигурационного файла.

Кроме декларации единого формата модели данных, стандарт FACE [5; 6, с. 63] определяет требование к обеспечению адаптации между собой различных способов передачи пакетов: Publish/Subscribe (Публикация/Подписка) и Request/Response (Запрос/Ответ), что также требует рассмотрения.

Результаты исследования и их обсуждение

В концепции FACE [6, с. 4–12] предлагается следующая архитектура:

− сегмент операционной системы (Operating System Segment – OSS),

− сегмент перемещаемых компонентов (Portable Components Segment – PCS),

− сегмент служб, специфичных для платформы (Platform-Specific Services Segment – PSSS),

− сегмент транспортных служб (Transport Services Segment – TSS),

− сегмент ввода/вывода (I/O Services Segment – I/O SS).

В настоящее время в российской авиационной промышленности отсутствует стандарт, аналогичный FACE, и разрабатываемые приложения учитывают только стандарт ARINC 653 [9; 10]. Поэтому предложено реализовать задачи преобразования информации от/для устройств ввода/вывода, решаемые в PSSS и TSS с помощью Адаптера [3; 8].

На данном этапе специальная программа-Адаптер берет на себя функции унификации и адаптации, что позволяет на данном этапе иметь адаптеру требуемый функционал без создания дополнительных программных средств.

В зависимости от задач, которые планируется решать с помощью Адаптера, необходимо формирование расписания вызова и определение порядка обслуживания портов. В частном случае, Адаптер, привязанный к конкретному приложению, может быть вызван непосредственно перед самим приложением для обработки его входящих сообщений.

Входные и выходные порты Адаптера увязываются с входными и выходными портами функциональных приложений, так же как входные порты одного функционального приложения и выходные порты другого увязываются через Адаптер.

Для апробации возможностей Адаптера решено встроить его в процесс информационного обмена между функциональным программным обеспечением «автопилота» (AFA), системой самолетовождения (Flight Management System – FMS) и внешним окружением [3; 8; 9]. Стоит отметить,

что Адаптер позволяет работать и с другими функциональными приложениями. Для этого необходимо произвести настройку конфигурационных файлов.

Программное обеспечение Адаптера реализовано для запуска под операционной системой реального времени (ОСРВ) JetOS [9; 11; 13], разработанной ФАУ «ГосНИИАС» и используемой для моделирования других компонентов.

При подготовке запуска информационного обмена с использованием адаптера необходимо включить в состав интеграционного проекта те модули, между которыми должен быть настроен обмен, а также подготовить и настроить конфигурационные и настроечные файлы.

Важной особенностью реализации является возможность реализовать наличие внешней среды, что является актуальным на этапе включения моделей в комплекс-демонстратор.

Для упрощения создания нового проекта информационного обмена на основе шаблона разработан скрипт, заполняющий изменяемые данные на основе пользовательского ввода и дающий дальнейшие инструкции по заполнению конфигурационных и настроечных файлов.

Процесс работы скрипта в окне терминала представлен на рис. 1. Настройка производится в соответствии с рекомендационными материалами, доступными пользователям.

В частности, вводятся имена взаимодействующих модулей (приложения, участвующие в обмене), ограничения на размер сообщения без очереди, длительность, задается файл с заголовками функций трансформации, добавляется файл с определением функций трансформации, где функции соответствуют определенным сигнатурам, подключается библиотека поддержки функций, обновляется, если необходимо, атрибут процесса «Период».

При запуске такой конфигурации информационного взаимодействия функционального программного обеспечения системы FMS и системы AFA в терминале может отслеживаться процесс информационного обмена.

Пример отладочной печати приведен на рис. 2.

Дальнейшим развитием данного решения является его работа в рамках демонстратора информационного обмена, где устанавливается взаимодействие с внешней средой. При этом оценивается корректность работы Адаптера и приложений, запущенных в операционной системе реального времени JetOS.

```
user@isp-vm:/media/sf_CommonFolder/other-works/TSS-project$ ./tss_launcher

*****************************************************
*              TRANSPORT SERVICES SEGMENT           *
*****************************************************
*     Введите 1 - Запустить существующий проект     *
*     Введите 2 - Очистить данные сборки и запуска  *
*     Введите 3 - Создать новый проект              *
*****************************************************

3
Введите имя нового проекта
AFA_FMS
Введите частоту [Гц] отправки сообщений каждым приложением
Возможно выбрать частоту 5, 10, 20, 40, 50 или 100 Гц
50
Введите имя первого приложения, участвующего в информационном обмене (P1):
AFA
Введите имя второго приложения, участвующего в информационном обмене (P2):
FMS
Введите максимальный (наибольший) размер сообщения в порту без очереди:
1024
Введите имя заголовочного файла (без расширения), который содержит объявление функций трансформации и описание
 типов передаваемых данных:
AFA_FMS.h
Новый проект создан. Все введенные данные учтены. Осталось добавить:
- Файлы с исходным кодом общающихся приложений
- Файлы с описанием передаваемых данных и функций трансформации
- Подробнее смотреть пункты 3.4 - 3.6 MANUAL.txt
Для запуска (после добавления файлов):
1) выполните повторно команду ./tss_launcher
2) выберите вариант под цифрой 1
3) затем укажите имя созданного проекта
user@isp-vm:/media/sf_CommonFolder/other-works/TSS-project$
```

Рис. 1. Интерфейс создания нового проекта информационного обмена

```
COM0,single,USR>
COM0,single,USR> ----------------------------------_start_connection_----------------------------------
COM0,single,USR> stubAFA: sending msg:
COM0,single,USR> stubAFA: HDG_TRK_ref    = 4086.000000
COM0,single,USR> stubAFA: ALT_ref        = 2749.000000
COM0,single,USR> stubAFA: VS_ref         = 12767.000000
COM0,single,USR> stubAFA: FPA_ref        = 9084.000000
COM0,single,USR> stubAFA: AT_THR_TGT_L   = 0.000000
COM0,single,USR> stubAFA: AT_THR_TGT_L_V = 0
COM0,single,USR> stubAFA: AT_THR_TGT_R   = 25089.000000
COM0,single,USR> stubAFA: AT_THR_TGT_R_V = 1
COM0,single,USR> stubAFA: FGCP_SPD       = 26966.000000
COM0,single,USR> stubAFA: FGCP_MACH      = 4978.000000
COM0,single,USR> stubAFA: FGCP_HDG       = 20495.000000
COM0,single,USR> stubAFA: FGCP_ALT       = 10311.000000
COM0,single,USR> stubAFA: FGCP_VS        = 11367.000000
COM0,single,USR> stubAFA: FGCP_FPA       = 30054.000000
COM0,single,USR> stubAFA: AP_DSCR_1      = 00000000000000000001001000000000
COM0,single,USR> stubAFA: AP_DSCR_2      = 00000000000000000010000000000101
COM0,single,USR> stubAFA: AT_DSCR_1      = 00000000000000000000000000000110
COM0,single,USR>
COM0,single,USR>
COM0,single,USR> stubAFA: idOutputPort1 new message send
COM0,single,USR> TS5: INPUT_THERE_PORT new message read and will transform and send in OUTPUT_THERE_PORT
COM0,single,USR> TS5: new message transformed
COM0,single,USR> TS5: OUTPUT_THERE_PORT new message send
COM0,single,USR> stubFMS: new msg was read:
COM0,single,USR> stubFMS: AP_HDG_TGT    = 4086.000000
COM0,single,USR> stubFMS: AP_ALT_TGT    = 2749.000000
COM0,single,USR> stubFMS: AP_VS_TGT     = 12767.000000
COM0,single,USR> stubFMS: AP_FPA_TGT    = 9084.000000
COM0,single,USR> stubFMS: AP_THR_TGT_01 = 0.000000
COM0,single,USR> stubFMS: AP_THR_DEF_01 = 0
COM0,single,USR> stubFMS: AP_THR_TGT_10 = 25089.000000
COM0,single,USR> stubFMS: AP_THR_DEF_10 = 1
COM0,single,USR> stubFMS: AP_CAS_TGT    = 26966.000000
COM0,single,USR> stubFMS: AP_MACH_TGT   = 4978.000000
COM0,single,USR> stubFMS: AP_TRK_TGT    = 20495.000000
COM0,single,USR> stubFMS: W_302         = 00000000000000101000000000000000
COM0,single,USR> stubFMS: W_303         = 00000000000000100110101000000000
COM0,single,USR>
COM0,single,USR>
COM0,single,USR> stubFMS: sending msg:
COM0,single,USR> stubFMS: CAS_CMD        = 14535.500000
COM0,single,USR> stubFMS: MACH_CMD       = 10668.000000
COM0,single,USR> stubFMS: ROLL_CMD       = 0.000000
COM0,single,USR> stubFMS: VgpDev         = 0.000000
COM0,single,USR> stubFMS: ALT_CMD        = 0.000000
COM0,single,USR> stubFMS: FPA_CMD        = 1590.500000
COM0,single,USR> stubFMS: TRUE_RQST      = 1
COM0,single,USR> stubFMS: VERT_GUID_READY = 1
COM0,single,USR> stubFMS: LAT_GUID_READY  = 1
COM0,single,USR> stubFMS: CAS_RQST       = 0
COM0,single,USR> stubFMS: FPA_RQST       = 1
COM0,single,USR> stubFMS: SPD_THR_RQST   = 0
COM0,single,USR> stubFMS: ALT_RQST       = 1
COM0,single,USR> stubFMS: LANDING_RQST   = 1
COM0,single,USR> stubFMS: HDG_RQST       = 0
COM0,single,USR> stubFMS: MACH_RQST      = 1
COM0,single,USR> stubFMS: ROLL_RQST      = 0
COM0,single,USR>
COM0,single,USR>
COM0,single,USR> stubFMS: idOutputPort1 new message send
COM0,single,USR> stubAFA: preparing to read msg
COM0,single,USR> TS5: INPUT_BACK_PORT new message read and will transform and send in OUTPUT_BACK_PORT
COM0,single,USR> TS5: new message transformed
COM0,single,USR> TS5: OUTPUT_BACK_PORT new message send
COM0,single,USR> stubAFA: preparing to read msg
COM0,single,USR> stubAFA: new msg was read:
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_103_SelectedAirspeed       = 14535.500000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_106_1_TargetMach          = 10668.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_121_1_HorizontalCommand   = 0.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_174_1_ILS_GLIDE_DEV       = 0.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_102_1_SelectedAltitude    = 0.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_322_1_TargetFlightPathAngle = 1590.500000
COM0,single,USR> stubAFA: FMS_DSCR_270 = 00000000000000110100000000000000
COM0,single,USR> stubAFA: FMS_DSCR_371 = 00000000000000000000011000000000
COM0,single,USR> stubAFA: FMS_DSCR_V   = 00001111000000000000000000001100
COM0,single,USR> ----------------------------------_end_connection_----------------------------------
COM0,single,USR>
```

Рис. 2. Пример отладочной печати информационного обмена

Заключение

В работе рассмотрен подход к унификации формата сообщений, которыми обмениваются авиационные функциональные приложения.

В рамках данного проекта:

1. Обоснована необходимость разработки адаптера информационного обмена.

2. Определена среда реализации проекта.

3. Рассмотрены сообщения, передаваемые моделями модулей системы FMS и системы AFA.

4. Разработаны форматы конфигурационных и настроечных файлов.

5. Создан адаптер, обеспечивающий:

− автоматическую транспортировку сообщений между приложениями;

− автоматическое преобразование содержимого сообщений;

− наличие функций, облегчающих реализацию преобразования.

6. Апробирован адаптер для обмена сообщениями между системами FMS и AFA.

7. Адаптер включен в состав демонстратора информационного обмена унифицированных программных компонентов на базе операционной системы JetOS.

Список литературы

1. Квалификационные требования. Часть 178С. Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники. Введено в действие с 22.11.2016. М.: Авиарегистр МАК, 2016. 105 с.

2. Batuwangala E., Kistan T., Gardi A., Sabatini R. Certification challenges for next-generation avionics and air traffic management systems. IEEE Aerospace and Electronic Systems Magazine. 2018. Vol. 33. No. 9. P. 44–53

3. Naushad F., Dehriya S. Unified Interconnect System for Next Generation Fighter Aircraft. InternatIonal Journal of electronics and communication technology. 2015. Vol. 6. No. 3. P. 95–100

4. Кравченко О. Обмен сообщениями – стремление к унификации // Компьютер-пресс. 2011. № 11. URL: https://compress.ru/article.aspx?id=12262&ysclid=l9n6vm5gbf983054939 (дата обращения: 05.10.2022).

5. The Open Group FACE™ Consortium. FACE FAQs. [Электронный ресурс]. URL: https://www.opengroup.org/content/future-airborne-capability-environment-face/faqs (дата обращения: 05.10.2022).

6. Technical Standard for Future Airborne Capability Environment (FACE™). Edition 3.0. The Open Group. 2017. 569 p.

7. Буздалов Д.В., Зеленов С.В., Корныхин Е.В., Петренко А.К., Страх А.В., Угненко А.А., Хорошилов А.В. Инструментальные средства проектирования систем интегрированной модульной авионики // Труды ИСП РАН. 2014. [Электронный ресурс]. URL: https://www.ispras.ru/proceedings/docs/2014/26/1/isp_26_2014_1_201.pdf (дата обращения: 05.10.2022).

8. Книга В.Е., Жаринов И.О., Богданов А.В., Виноградов П.С. Принципы организации архитектуры перспективных бортовых цифровых вычислительных систем вавионике // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 2 (84). [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/printsipy-organizatsii-arhitektury-perspektivnyh-bortovyh-tsifrovyh-vychislitelnyh-sistem-v-avionike/viewer (дата обращения: 05.10.2022).

9. Маллачиев К.М., Пакулин Н.В., Хорошилов А.В. Устройство и архитектура операционной системы реального времени // Труды ИСП РАН. 2016. № 28 (2). С. 181–192.

10. ARINC Specification 653 Avionics Application Software Standard Interface Set. SAE-ITC. 2019. Vol. 1. No. 5. 118 p.

11. ГОСНИИАС.2200.100.1388-001СП. Спецификация требований к сервисному ПО, обеспечивающему информационный обмен унифицированных функциональных модулей. М.: ГосНИИАС, 2020. 18 с.

12. Молев А.А. Метод автоматического формирования телекоммуникационных модулей структурных элементов автоматизированных систем на основе XML-описания // Программные и аппаратные средства. 2017. № 1. [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/metod-avtomaticheskogo-formirovaniya-telekommunikatsionnyh-moduley-strukturnyh-elementov-avtomatizirovannyh-sistem-na-osnove-xml/viewer (дата обращения: 05.10.2022).

13. Инструкция по эксплуатации (JetOS Manual Guide). Ревизия 0.53. М.: ГосНИИАС, ИСП РАН, 2022. 119 с.

14. IEEE 754 Standard for Floating-Point Arithmetic. IEEE Std. 2008. 70 p.