

DOI: 10.15514/ISPRAS-2024-36(5)-12



Архитектура открытого программного комплекса УЕМКА для управления целевыми устройствами SMART-наноспутников

Г.А. Щеглов, ORCID: 0000-0002-0129-0439 <shcheglov_ga@bmstu.ru>
К.А. Жданова, ORCID: 0009-0000-6830-7808 <k.a.zhdanova@yandex.ru>
З.С. Жумаев, ORCID: 0000-0002-5310-0182 <zhumayev@bmstu.ru>
Н.Д. Каменев, ORCID: 0009-0000-9029-2681 <physicorym@gmail.com>

Московский государственный технический университет имени Н.Э. Баумана,
105005, Россия, г. Москва, ул. 2-я Бауманская, д. 5, к. 1.

Аннотация. В работе показано, что актуальной проблемой при разработке наноспутников является отсутствие открытых программных средств для бортовых вычислительных устройств и «умной» полезной нагрузки. Рассматривается разработка открытого программного комплекса для централизованного управления целевыми конечными устройствами наноспутников на базе микросервисной архитектуры. Показаны преимущества использования данного подхода при создании программного комплекса. Предложено использование имитационной модели наноспутника для оперативной отладки и тестирования программного комплекса. Авторами работы приведена структура программного комплекса и показано место имитационной модели в ней. Работа является развернутым обзором разработанного авторами программного комплекса УЕМКА.

Ключевые слова: наноспутник; класс наноспутников CubeSat; программно-определяемые устройства; микросервис; контейнеризация; имитационная модель; виртуализация; нейронная сеть; клиент-серверное программирование; сервис-ориентированная архитектура.

Для цитирования: Щеглов Г.А., Жданова К.А., Жумаев З.С., Каменев Н.Д. Архитектура открытого программного комплекса УЕМКА для управления целевыми устройствами SMART-наноспутников. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 163–180. DOI: 10.15514/ISPRAS–2024–36(5)–12.

Благодарности: Работа выполнена при поддержке Фонда содействия инновациям (ФСИ) группой авторов, сотрудников МГТУ им. Н.Э. Баумана, организовавших стартап ООО «Космические вычислительные системы» (№24 ГКод-ЦТ-С17-Д5/82177 от 27.12.2022) в рамках конкурса «Код-Цифровые технологии» (очередь II) в рамках федерального проекта «Цифровые технологии» национальной программы «Цифровая экономика Российской Федерации».

Open-Source Software Architecture UEMKA for Controlling SMART-Nanosatellite Target Devices

G.A. Shcheglov, ORCID: 0000-0002-0129-0439 <shcheglov_ga@bmstu.ru>

K.A. Zhdanova, ORCID: 0009-0000-6830-7808 <k.a.zhdanova@yandex.ru>

Z.S. Zhumaev, ORCID: 0000-0002-5310-0182 <zhumaev@bmstu.ru>

N.D. Kamenev, ORCID: 0009-0000-9029-2681 <physicorym@gmail.com>

*Bauman Moscow State Technical University,
105005, Moscow, 2-nd Baumanskaya st., 5.*

Abstract. The paper shows that a pressing problem in the development of nanosatellites is the lack of open software for on-board computing devices and “smart” payloads. The development of an open software package for centralized management of target terminal devices of nanosatellites based on microservice architecture is considered. The advantages of using this approach when creating a software package are shown. It is proposed to use a nanosatellite simulation model for operational debugging and testing of the software package. The authors of the work present the structure of the software complex and show the place of the simulation model in it. The work is a detailed review of the UEMKA software package developed by the authors.

Keywords: nanosatellite; CubeSat; software defined devices; microservice; containerization; simulation model; virtualization; artificial neural network; client-server programming; service-oriented architecture.

For citation: Shcheglov G.A., Zhdanova K.A., Zhumaev Z.S., Kamenev N.D. Open-source software architecture UEMKA for controlling SMART-nanosatellite target devices. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024. pp. 163-180 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-12.

Acknowledgements. The work carried out with the support of the Innovation Promotion Fund (IPF) by a group of authors, employees of the N.E. Bauman Moscow State Technical University, who organized the startup Space Computing Systems LLC (No. 24 GKod-CT-C17-D5/82177 dated 12/27/2022) within the framework of the Code-Digital Technologies competition (Phase II) within the framework of the federal project Digital technologies” of the national program “Digital Economy of the Russian Federation”.

1. Введение

Облик перспективных космических аппаратов определяется технологиями четвертой промышленной революции и, прежде всего, принципами киберфизики: взаимодействием физических объектов с информационными полями. Эти принципы порождают в зарубежных (прежде всего американских) проектах принципиально новые архитектуры космических аппаратов, построенные на базе распределенных информационных систем с высокой степенью виртуализации: SMART-satellites – «умные» спутники [1-3]. Подобный подход обеспечивает высокую серийность производства спутников и построение интегрированных в единое информационное пространство много- и мега- спутниковых группировок (т.н. созвездий). Массовое производство спутников является актуальной задачей развития российской космической отрасли [4].

Материалы международных конгрессов по астронавтике (IAC), показывают актуальность внедрения в спутниковые информационные системы повсеместных вычислений (ubiquitous computing), как облачных (cloud computing) [5], так и граничных (edge computing) [6]. Показателен пример развития информационных технологий четвертого поколения, применяемых в КНР для разработки и тестирования бортовых информационных систем на базе цифровых двойников [7]. Подобные подходы применяются единообразно для спутников различного класса, в том числе наноспутников.

Бортовое программное обеспечение является основой SMART-спутников. Важным свойством SMART-спутников является возможность гибкого управления их программным обеспечением на орбите. Успешность выполнения космических миссий наноспутников во

многим определяется качественной разработкой программного обеспечения для управления вычислительными устройствами.

Программно-математическое обеспечение (ПМО) для управления с Земли SMART-вычислительными устройствами наноспутника имеет закрытый программный код. Данное обстоятельство препятствует разработке созвездий, кластеров и группировок спутников сообществом разработчиков, что ограничивает эффективность применения созвездий наноспутников в космосе.

Обзор имеющихся технологий показал [8-11], что в области бортового ПМО искусственных спутников острее всего не хватает средств удаленного администрирования прикладных программ, выполняемых на бортовых компьютерах для обработки изображений и сигналов, в том числе с использованием параллельных вычислений и нейросетей. Такое обеспечение относится к категории Unified Endpoint Management (UEM).

Целью данной работы является описание созданного открытого программного комплекса UEM для космических аппаратов (КА) – наноспутников класса CubeSat (UEMKA), который позволяет осуществлять удаленное управление космическими вычислительными устройствами с Земли и будет единым для разработчиков конечных устройств. Это позволит гибко встраивать различные полезные нагрузки на борт наноспутников и управлять ими единым образом с помощью вычислительных устройств. В работе особое внимание уделяется архитектуре программного комплекса и вопросам объединения наземного и космического сегментов в единую информационную среду.

2. SMART-наноспутники: современные программно-определяемые космические аппараты

Современные технологии, относящиеся к четвертой промышленной революции, меняют проектный облик космических аппаратов. Происходит повышение управляемости спутников: они получают в названии приставки «smart» (умный, сообразительный). После смартфонов, смарт-часов, смарт-очков востребованность получили SMART-satellites – «умные» спутники, как показано на рис. 1. Такие спутники, фактически представляющие собой летающий в космосе компьютер, разрабатывает, в частности, фирма Lockheed Martin (США) в рамках проекта SmartSat technology. Как и для других SMART-устройств, для SMART-спутников характерна миниатюризация: они относятся к категории наноспутников массой до 10 кг. Например, спутник In-space Upgrade Satellite System (LINUSS) выполнен по стандарту CubeSat в форм-факторе 12U [12].

Известные архитектуры открытого бортового программного обеспечения для спутников [13] и беспилотных летательных аппаратов [14-16] рассчитаны на использование программируемых контроллеров в бортовых информационных системах, которые ограничены по функционалу. Создание космических киберфизических систем связано с переносом существующих технологий и архитектур, активно применяемых в наземных информационных системах, в космические информационные системы [17].

Программно-определяемые спутники, или SMART-спутники, представляют собой новое поколение космических аппаратов, обладающих возможностью гибкой настройки и программирования функций непосредственно на орбите. Это позволяет им адаптироваться к изменяющимся потребностям и условиям в реальном времени, что делает их более эффективными и универсальными в сравнении с традиционными спутниками. Ряд компаний, таких как, SpaceX, Planet Labs, OneWeb, Spire Global, разрабатывают и запускают свои уникальные спутники с использованием собственных технологий и программного обеспечения. В табл. 1 показаны некоторые программно-определяемые спутники, выведенные на орбиту.



Рис. 1. Основные отличительные особенности SMART-спутника.
 Fig. 1. Main distinctive features of the SMART-satellite.

Табл. 1. Обзор программно-определяемых спутниковых платформ.
 Table 1. Overview of Software Defined Satellite Platforms.

№	Наименование	Разработчик	Страна	Дата запуска	Космическая миссия
1	Группировка Dove (CubeSat 3U)	Planet Labs, Inc	США	24.01.18-н.в.	ДЗЗ, обнаружение природных бедствий, спутниковый мониторинг в реальном времени
2	Eutelsat 10B	Thales Alenia Space	Франция, Италия	23.11.22	Предоставление мобильной, морской и авиационной связи, предоставление видеослужб
3	Группировка Iridium NEXT	Thales Alenia Space, Orbital ATK	США	14.01.17-11.01.19	Высокоскоростная мобильная связь, поддержание сервисов IoT
4	Группировка Sentinel-2	Airbus Defence and Space	Франция	23.06.15-7.03.17	ДЗЗ, мониторинг изменчивости условий земной поверхности
5	Tianxing-1	Innovation Academy for Microsatellites of the Chinese Academy of Sciences	Китай	22.06.22	Экспериментальный спутник для изучения околоземного космического пространства
6	Группировка LM LINUSS SmartSat (CubeSat 12U)	Lockheed Martin	США	18.04.23	Отработка программно-определяемой архитектуры спутника
7	Connecta T1.2 (CubeSat 3U)	Plan-S	Турция	03.01.23	Испытания двунаправленной связи IoT
8	Ovzon 3	Maxar Technologies	Швеция	03.01.24	Спутник L-диапазона для обеспечения мобильной широкополосной связи в недостаточно обслуживаемых регионах

SMART-спутники оснащены мощными компьютерами и программным обеспечением, которое позволяет им выполнять широкий спектр задач. Эти задачи могут быть решены с использованием математической обработки данных непосредственно на борту аппарата.

Они также используются для создания информационных сетей спутников (Non-terrestrial networks, NTN), обеспечивая возможность совместной работы и обмена данными между собой и потребителями в рамках интернета вещей (Internet of Things, IoT). Например, основная особенность группировки спутников Dove состоит в использовании высокоскоростной радиосвязи для оперативной передачи данных на наземные радиостанции. С помощью высокоскоростной нисходящей линии связи X-диапазона на низкой околоземной орбите возможно передавать до 1Тб данных с одного спутника на одну наземную станцию за 24 часа [18]. Многоцелевой спутник Eutelsat 10B – спутник связи с высокой пропускной способностью порядка 35 Гбит/с. Вся полезная нагрузка спутника обрабатывается в цифровом виде, что обеспечивает гибкость распределения мощности благодаря цифровому процессору [19].

Одним из основных преимуществ SMART-спутников является возможность их эволюции по назначению. В частности, в области дистанционного зондирования Земли (ДЗЗ), возможность обработки изображений на борту спутника при помощи нейронных сетей позволяет менять целевую задачу путем замены модели: например контроль за пожарами может быть заменен на контроль за наводнениями и пр. Благодаря возможности программной настройки, SMART-спутники могут быть более экономичными в эксплуатации, поскольку их функциональность адаптирована под конкретные задачи без необходимости запуска новых спутников. Это также способствует сокращению времени и затрат на разработку и запуск космических аппаратов.

Если полезная нагрузка наноспутников, применявшихся в последнее десятилетие, была построена на базе программируемых контроллеров, то полезные нагрузки SMART-наноспутников строятся на основе микрокомпьютеров и локальных вычислительных сетей. Возрастает располагаемая вычислительная мощность бортовой аппаратуры.

Вычислительные устройства представляют SMART-блоки бортовой аппаратуры научно-образовательного наноспутника класса CubeSat. Примером такого SMART-блока является бортовой вычислительный модуль, реализованный на базе одноплатного многоядерного микрокомпьютера и предназначенный для проведения орбитальных вычислительных экспериментов [20]. На рис. 2 показана летная модель такого модуля, интегрированная на борт наноспутников Ярило 3 и Ярило 4, разработанных в МГТУ им. Н.Э. Баумана и запущенных на низкую околоземную орбиту 27 июня 2023 года.

Подобные блоки аппаратуры широко применяются, в частности, для обработки фотографий, сделанных бортовыми камерами, обработки результатов измерений, полученных датчиками и целевой аппаратурой, проведения высокопроизводительных вычислений, необходимых для перехода от передачи на Землю «сырых» данных к передаче «знаний», повышающего эффективность выполнения программы полета.

Современные наноспутники содержат на борту как физические SMART-устройства, представляющие собой бортовые микрокомпьютеры и вычислительные модули, так и виртуальные SMART-устройства, представляющие собой изолированные программные модули, виртуальные машины и контейнеры.

В проекте UEMKA в качестве виртуальных SMART-устройств рассматриваются программы-клиенты и программы-серверы, запущенные в Docker-контейнерах, а в качестве макетов SMART-устройств выступают одноплатные микрокомпьютеры типа Raspberry Pi 3 и 4 с CAN-шиной (Controller Area Network), необходимой для работы устройства в составе КА.

Макет SMART-устройства состоит из микрокомпьютера Raspberry Pi и модуля CAN Bus на базе микросхемы MCP2515 для обеспечения физической шины передачи данных. Объединение нескольких макетов SMART-устройств между собой имитирует работу сети

бортовых вычислительных устройств на борту наноспутника. Подключение SMART-устройства к виртуальной шине (ноутбуку) происходит с помощью модуля преобразования USB-CAN.



*Рис. 2. Летная модель бортового вычислительного модуля.
Fig. 2. Flight model of the on-board computing module.*

Подробное описание технических требований для работы с программным комплексом UEMKA, а также первичная настройка микрокомпьютера и подключение макетов представлено в разделе документации и Wiki репозитория проекта UEMKA [21].

3. Принципы построения архитектуры программного комплекса

Архитектура программного комплекса UEMKA, обеспечивающая централизованное управление целевыми конечными устройствами наноспутников в рамках единой информационной системы «Земля–Борт», создана с учетом современных требований к гибким распределенным информационным системам:

- Архитектура системы является открытой, поскольку проект UEMKA реализован как программное обеспечение с открытым исходным кодом.
- Архитектура системы ориентирована на сервисы (SOA – service-oriented architecture) и состоит из клиентских и серверных приложений.
- Архитектура системы UEMKA обеспечивает разделение функций между ее элементами, что выдвигает требование к созданию многоуровневой архитектуры.
- Архитектура системы UEMKA обеспечивает эффективное управление связями между ее элементами на всех этапах жизненного цикла бортового программного обеспечения, что выдвигает требование к созданию программно-определяемой сегментированной архитектуры, основанной на технологиях виртуализации.

Контейнерная виртуализация позволяет комбинировать подпрограммы, зависящие от разных версий одних и тех же библиотек и даже от разных версий языков программирования. Современная контейнерная виртуализация предполагает системы управления контейнерами, например, систему Docker Compose, где имеются инструменты для отслеживания состояния и автоматического повторного запуска микросервисов, встроенные возможности по масштабированию. Использование подобных систем в проекте UEMKA позволяет реализовать подход «инфраструктура как код», что уменьшает время на повторную настройку инфраструктуры, например, серверов баз данных, обмена сообщениями, логирования и визуализации.

Информационная система, построенная при помощи программного комплекса UEMKA, является сегментированной, что подразумевает наличие в общей информационной системе отдельных подсистем – сегментов. Внутри одного сегмента каналы передачи данных имеют

относительно высокую пропускную способность, в то время как пропускная способность каналов связи между сегментами существенно ниже.

Сегментированная архитектура подразумевает использование технологий виртуальных сетей и обмена сообщениями между микросервисами на основе подписки. В настоящее время наибольшее распространение получили подходы, основанные на контейнерной виртуализации, в частности по технологии Docker [22]. При таком подходе можно зафиксировать все зависимости в контейнере, что гарантирует воспроизводимость окружения и результатов работы программы. Воспроизводимость результатов особенно важна для космической техники.

В настоящее время подразумевается наличие как минимум двух сегментов: наземного (развернут на наземном комплексе управления) и космического (развернут на борту КА). Каждый сегмент построен как система клиентских (уровень сервисов) и серверных (уровень управления) приложений-микросервисов, работающих под управлением операционной системы с поддержкой контейнеризации.

Архитектура программного комплекса УЕМКА является открытой, что проявляется в реализации следующих принципов:

- открытость исходного кода, который размещен в сети Интернет на базе открытого репозитория и системы документирования кода; при работе над кодом использованы только программные модули с открытым исходным кодом или лицензиями, допускающими свободное использование кода на гетерогенной сети из компьютеров и одноплатных микрокомпьютеров;
- открытость состава программных модулей, которая предполагает, что реализованная в настоящее время номенклатура модулей является первоначальной и будет расширяться по мере развития проекта и формирования сообщества пользователей;
- открытость топологии связей между модулями, которая предполагает возможность гибкой настройки каналов обмена сообщениями между модулями путем формирования конфигурационных файлов;
- открытость функциональных возможностей комплекса, которая предполагает возможности масштабирования и эволюции по назначению бортовой информационной системы, построенной на основе данного комплекса; в частности, информационная система позволяет проводить по единой технологии разработку, тестирование, развертывание, эксплуатацию и модернизацию информационной системы КА в целом и ее отдельных элементов.

Удовлетворяющая указанным требованиям архитектура удобна для разработки бортового программного обеспечения быстро меняющимися коллективами, что особенно актуально для университетов, где над разработкой программ работают студенты и аспиранты.

4. Структура программного комплекса УЕМКА

Программный комплекс УЕМКА использует сервис-ориентированную архитектуру SOA, реализованную на базе микросервисов – малых автономных (как правило, развертываемых в контейнерах) программ, взаимодействующих посредством обмена сообщениями или при помощи выполнения сценариев командной строки операционной системы. Определяющим для проекта УЕМКА является то, что использование микросервисов дает существенные преимущества в автоматизированном развертывании и управлении информационной системой [23]. Состав программного комплекса УЕМКА показан на рис. 3.



Рис. 3. Состав программного комплекса UEMKA.
 Fig. 3. Composition of the UEMKA software package.

Для реализации программного комплекса используется среда разработки Microsoft Visual Studio Code, операционные системы Ubuntu версий 18.04, 20.04, 22.04 и Raspbian GNU/Linux версии 9, языки программирования C++ (компилятор g++ 7.5.0) и Python версий 3.6 и 3.10, система контроля версий Git, набор утилит can-utils [24] для работы с шиной CAN, система контейнерной виртуализации приложений docker и docker compose, базы данных PostgreSQL и SQLite, а также система визуализации данных Grafana.

Архитектура программного комплекса UEMKA является многоуровневой и включает:

- Уровень сервисов – верхний уровень, включающий программы–клиенты, реализующие логику решения целевых задач. Разработана библиотека специализированных программ–клиентов для управления бортовыми конечными устройствами.
- Уровень управления – уровень программ, работающих в режиме клиент–сервер и предназначенных для управления работой программ–приложений. Разработана библиотека программ–серверов для подготовки, развертывания приложений и

управления ими.

- Уровень данных – уровень электронных документов, используемых для передачи сообщений в процессе обмена информацией между программами. Средства уровня данных позволяют настраивать структуру реквизитной и содержательной частей электронных документов.
- Уровень шины – нижний уровень, включающий протокол передачи данных, средства программирования сокетов сети, средства организации каналов передачи данных.

Уровень данных и уровень шины образуют средства транспортного уровня и предназначены для организации удаленного доступа к бортовым конечным устройствам с единой централизованной управляющей консоли.

5. Особенности программного комплекса УЕМКА

5.1 Имитационная цифровая модель космического аппарата и внешней среды

В программном комплексе УЕМКА используется программно-аппаратное моделирование (Hardware in the loop), при котором системы наноспутника и влияние внешней среды моделируются численно на компьютере, а SMART-устройства являются реальными физическими объектами. Это позволяет отлаживать программное обеспечение SMART-блоков независимо от наноспутника.

Численное моделирование представлено имитационной цифровой моделью космического аппарата и внешней среды. Связанная математическая модель имитирует работу с реальным спутником, значительно упрощая процесс отладки и тестирования программного комплекса. На рис. 4 показана блок-схема математической модели космического аппарата. Математическая модель учитывает взаимовлияние системы ориентации и стабилизации (СОС) с моделями положения Солнца и магнитного поля Земли, тепловой двигательной установки (ДУ) и системы энергоснабжения (СЭП).

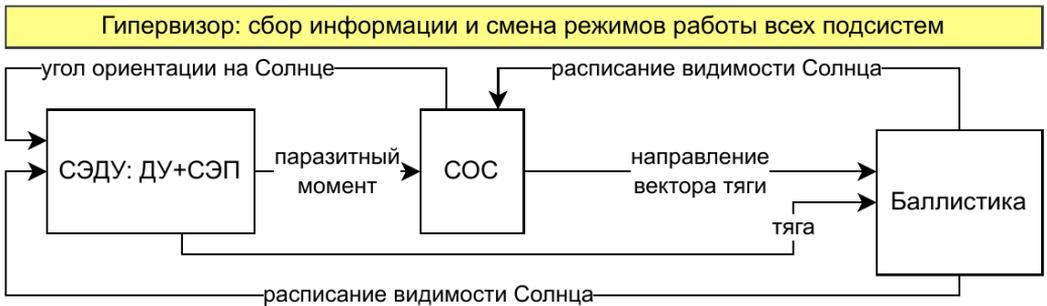


Рис. 4. Блок-схема математической модели космического аппарата.

Fig. 4. Spacecraft mathematical model diagram.

Общий вектор состояния спутника состоит из 20 компонент, 19 из которых независимые, так как скалярная часть кватерниона используется для проверки ошибки численного интегрирования из условия нормированности кватерниона:

$$y = \left(E, m_{fuel}, r_x, r_y, r_z, v_x, v_y, v_z, \omega_{bx}^{bi}, \omega_{by}^{bi}, \omega_{bz}^{bi}, q_0^{bi}, q_1^{bi}, q_2^{bi}, q_3^{bi}, \omega_{wh0}, \omega_{wh1}, \omega_{wh2}, \omega_{wh3}, Q_e \right)^T; \quad (1)$$

где E – общая тепловая энергия рабочего тела и конструкции ДУ, m_{fuel} – масса рабочего тела, $r_x \dots r_z, v_x \dots v_z$ – положение и скорость КА, $\omega_{bx}^{bi} \dots \omega_{bz}^{bi}$ – компоненты угловой скорости, $q_0^{bi} \dots q_3^{bi}$

– компоненты кватерниона ориентации, $\omega_{wh0} \dots \omega_{wh3}$ – угловые скорости маховиков, Q_e – заряд аккумуляторной батареи (АБ).

Накопленная тепловая энергия и суммарная масса двух фаз рабочего тела полностью определяют все термодинамические параметры, а ориентация спутника и заряд АБ определяют состояние СЭП. Подробное описание математической модели можно найти в диссертации [25] и в папке `satellite_model` репозитория [26].

Программный код по каждой системе наноспутника, упомянутой выше, представлен в репозитории в виде библиотеки функций [27]. В репозитории представлен также учёт несферичности поля тяготения Земли, модель магнитного поля IGRF и набор необходимых функций для демпфирования угловой скорости КА с помощью магнитных катушек по алгоритму B-Dot.

Архитектура программного комплекса численного моделирования внешней среды и космического аппарата разделена на две составляющие: модель бортовой электронно-вычислительной машины (БЭВМ) и модель остальных подсистем КА вместе с моделью внешней среды. Цикл моделирования разомкнутый: при достижении очередной точки останова выполняется запрос следующей задачи по шине CAN от БЭВМ.

Запустить моделирование можно как с использованием связки ПК и Raspberry Pi или все на ПК в Docker Compose. Второй вариант необходим для ускорения отладки и возможности проверки архитектуры работы системы с использованием двух шин данных с двусторонней трансляцией всех сообщений между шинами. Оба варианта показаны на рис. 5.

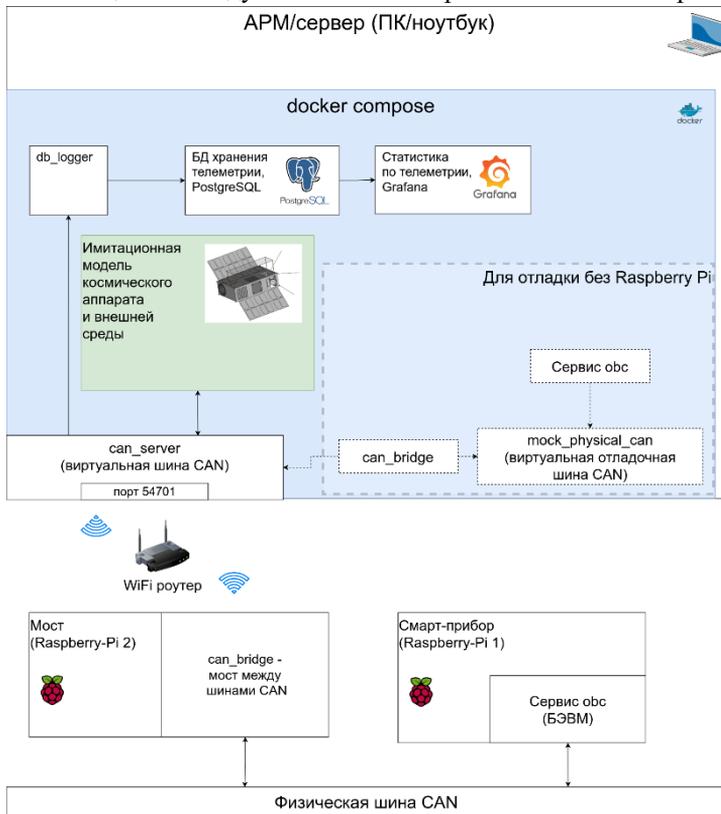


Рис. 5. Варианты запуска сервисов `obs`, `can_bridge`: основной на Raspberry Pi и отладочный в `docker compose`.
 Fig. 5. Options for launching `obs`, `can_bridge` services: main on Raspberry Pi and debugging in `docker compose`.

5.2 Применение нейронной сети для обработки космических снимков

SMART-устройства имеют обширный спектр целевых задач, в которые входит выполнение ресурсоемких вычислений мощностями микрокомпьютера; взаимодействие с полезной нагрузкой, подключенной к SMART-устройству; обработка результатов работы конкретных узлов спутника и т.п. В проекте UEMKA для реализации одной из целевых задач SMART-устройства используется бортовая камера для получения космических снимков и их анализа с помощью нейронной сети (НС).

При передаче изображений с бортовой камеры на Землю возникают проблемы с контекстом на кадрах. Изображения могут содержать различные виды шума, такие как облака или битые кадры. Для предотвращения передачи избыточной информации на Землю предлагается использовать НС, которая будет анализировать полученные снимки перед отправкой и определять содержимое изображений. Если обнаружено отсутствие шума, НС будет отправлять соответствующие изображения пользователю. Обработка изображений с использованием технологий искусственного интеллекта (ИИ) позволяет минимизировать передаваемые данные, обрабатывая их непосредственно на борту наноспутника.

Алгоритм работы НС показан на рис. 6.

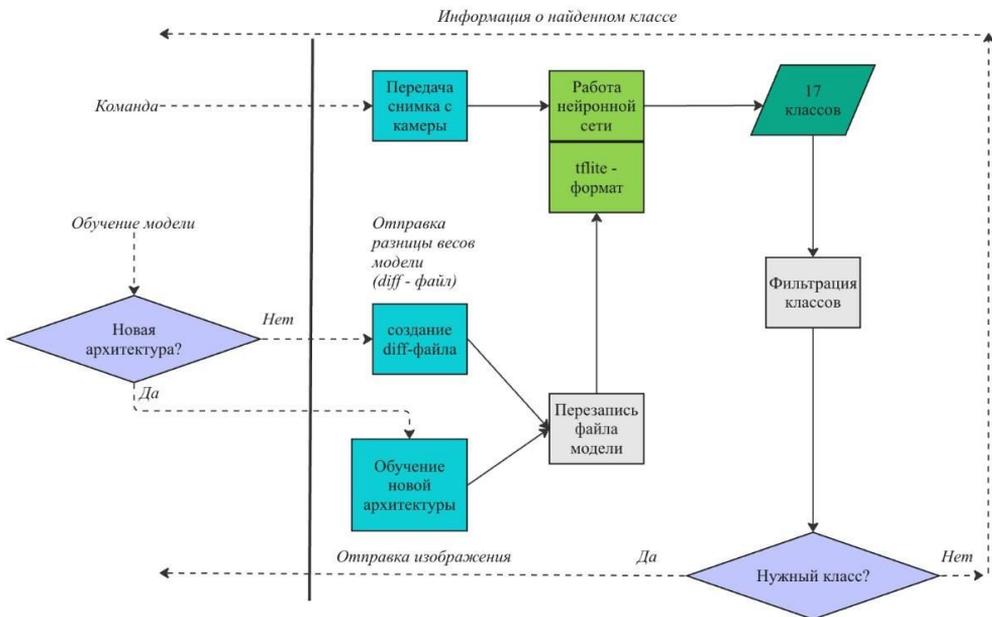


Рис. 6. Алгоритм работы НС.
Fig. 6. ANN operation algorithm.

При получении команды о создании снимка SMART-устройство передает сигнал бортовой камере. Готовое изображение на борту наноспутника обрабатывается НС на обнаружение целевых объектов, будь то вспышки на Солнце, конкретные звезды, лесные пожары и т.д. Затем обработанные снимки, соответствующие миссии КА, передаются наземному комплексу управления. Это значительно уменьшает количество «сырых» передаваемых данных и разгружает канал связи для иных задач. Программы для работы с НС представлены в репозитории проекта UEMKA [28].

Отличительной особенностью программного комплекса UEMKA является то, что в нем учтена возможность изменения летной программы КА, когда наноспутник уже находится на орбите. Для этого реализована удаленная настройка конфигураций SMART-устройств,

позволяющая изменить режим работы, например, ограничить целевую функцию НС, исключив из анализа конкретные объекты (облака, дороги и т.п.). Для этого на борт пересылается файл разности моделей (diff-файл).

Для космического радиоканала наноспутника размер передаваемого файла имеет ключевое значение. Размер модели около 15Мб, в то время как размер diff-файла измеряется в Килобайтах. Таким образом, согласно рис. 6, если для НС требуется новая архитектура, то на борт отправляется модель целиком, если же используется прежняя архитектура, которую дообучили на новых данных, то отправляется diff-файл. Выигрыш в размере файла очень важен при ограниченной скорости радиоканала и малой длительности сеанса связи с низкоорбитальным КА.

5.3 Графический интерфейс пользователя и система журналирования

Графический интерфейс пользователя (англ. GUI) предназначен для полуавтоматического и ручного управления удаленными полезными нагрузками наноспутников по шине передачи данных CAN. Эффект от использования графического интерфейса пользователя состоит в повышении удобства работы и уменьшении ошибок оператора наземного сегмента информационной системы полезных нагрузок SMART-спутников.

В проекте UEMKA используется веб-интерфейс с использованием фреймворков языка Python, который вместо разработки монолитного приложения позволяет использовать ресурсы браузера и дает возможность в будущем организовать удаленное управление полезными нагрузками через Интернет. Разработка интернет-технологий управления спутниковыми группировками через Интернет, с использованием мобильных устройств (телефонов, планшетов и пр.) в качестве терминалов, является мировым трендом [29].

Графический интерфейс решает следующие основные задачи:

- формирование структуры транспортного слоя;
- генерация электронных документов на уровне данных;
- управление клиентами-серверами UEMKA.

В состав интерфейса входят следующие компоненты.

- приложение PULT [30] с веб-интерфейсом, интегрированное с системой журналирования фреймов;
- приложение GUI [31] для формирования системы подписок на сообщения.

Приложение PULT, реализующее графический интерфейс, написано на языке Python с использованием фреймворка FastAPI, библиотеки SQLAlchemy для работы с реляционной базой данных SQLite. Структура веб-приложения состоит из системы веб-страниц как показано на рис. 7.

Приложение GUI предназначено для автоматизации процесса генерации и контроля файлов подписки.

Основными задачами приложения GUI являются:

- визуализация и управление составом системы;
- визуализация и управление структурой связей системы;
- генерация файлов подписки для всех элементов системы.

Для разработки GUI использован стек веб-технологий: гипертекстовая разметка страниц (html) и каскадных таблиц стилей (css), язык JavaScript, специализированный фреймворк React flow для создания графических интерфейсов на основе построения узлов, JavaScript-библиотека React.js для разработки графических интерфейсов и программная платформа Node.js для работы с языком JavaScript, предназначенная для создания серверных приложений.

В проекте УЕМКА также используется система журналирования информационного потока «Земля-борт», предназначенная для сохранения информационных пакетов, передаваемых по каналу «Земля-борт», в реляционную базу данных. Система журналирования реализована в виде базы данных фреймов, встроенной в приложение PULT, которая построена на базе фреймворка SQLAlchemy. Данный инструментариий предоставляет полный набор известных шаблонов сохранения данных на основе ORM-модели, разработанных для эффективного и высокопроизводительного доступа к базе данных, адаптированных к языку Python.

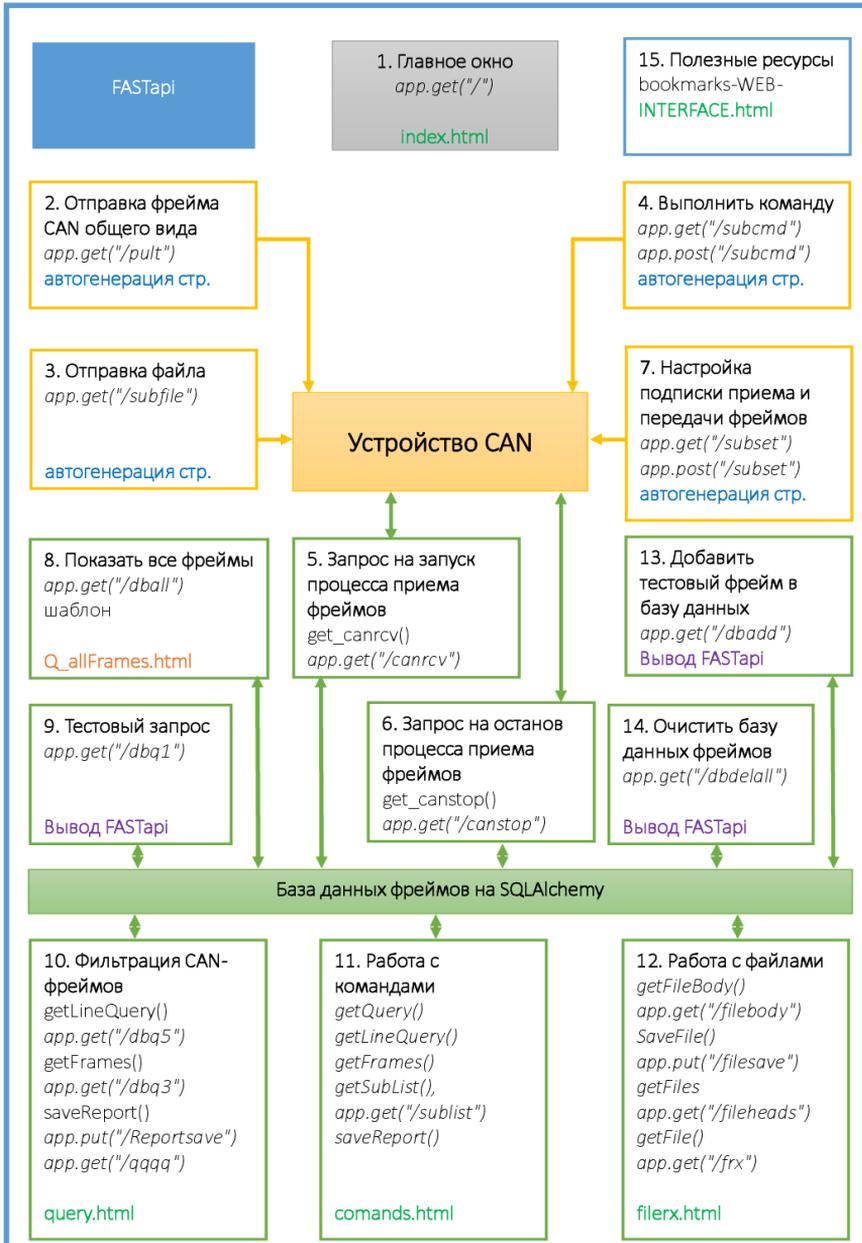


Рис. 7. Структурная схема графического интерфейса пользователя.
 Fig. 7. Block diagram of the graphical user interface.

6. Пример использования программного комплекса UEMKA

Рассмотрим информационную систему, созданную при помощи программного комплекса UEMKA, которая состоит из отдельных сегментов. Сегменты связаны каналами передачи данных по шине CAN. Каждый сегмент включает один или несколько сервисов, которые осуществляют обмен сообщениями и электронными документами по шине CAN. Каждый сервис включает как минимум программу-клиента для удаленного доступа к конечному устройству, а также программу-клиента, реализующего логику решения целевых задач (параллельные вычисления, работа с бортовой камерой, обработка изображений НС, изменение конфигураций и т.д.).

Для работы последовательно выполняется подготовка к развертыванию информационной системы (компиляция программ из исходных кодов, компиляция образов docker-контейнеров из Dockerfile), а затем и само развертывание (запуск транспортного уровня, запуск сервисов). Далее происходит управление сервисами путем передачи сообщений и файлов по CAN-шине и управление информационной системой (переконфигурирование путем запуска и останова контейнеров, изменения конфигурационных файлов и т.п.).

В рамках развертывания информационной системы «Земля–Борт» сегменты UEMKA могут быть развернуты единообразно в различных вариантах аппаратного обеспечения:

- на одном компьютере (ноутбуке);
- на гетерогенной сети из компьютеров и одноплатных микрокомпьютеров типа Raspberry Pi;
- в штатном режиме на аппаратном обеспечении центра управления полетом и бортовых вычислительных устройствах КА.

На рис. 8 показана простейшая информационная система, сегменты которой развернуты на одном компьютере. Цель данной системы – отладка средств комплекса UEMKA в процессе его разработки. В данном примере назначением системы является проведение тестовых параллельных вычислений в сегменте «Борт» под управлением сегмента «Земля». Как видно из схемы, сегмент «Борт» является виртуальным и запущен в контейнере с именем «SC_HPC». Сегмент «Земля» развернут вне контейнера в операционной системе. Транспортный уровень реализован при помощи виртуальных CAN-устройств. Управление удаленным доступом к конечному устройству – сервису для проведения параллельных вычислений программ-клиентом hpcst, реализовано при помощи микросервисов – программ-клиентов clt, работающих в различных режимах.

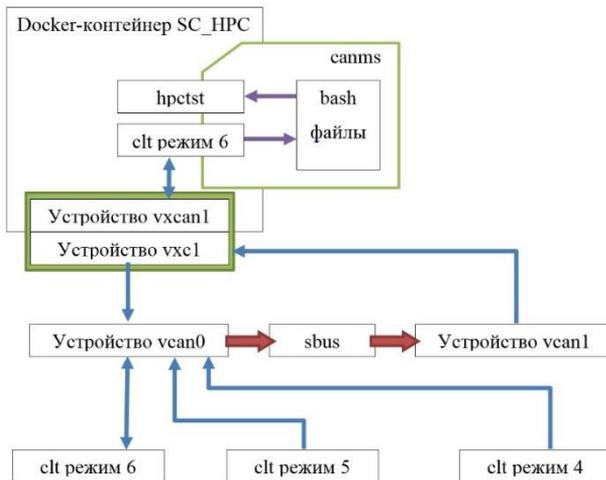


Рис. 8. Схема информационной системы.
Fig. 8. Information system diagram.

С помощью программного комплекса УЕМКА также реализуются другие сценарии работы информационной системы:

- распределенная информационная система «Земля–Борт»;
- работа с сервисом обработки изображений с помощью HC;
- работа с сервисом графического интерфейса пользователя;
- режим тестирования средств удаленного устранения ошибок и отказов SMART-устройств;
- режим тестирования программного комплекса на макетах SMART-устройств с применением камеры.

Примеры вариантов использования кода открытой библиотеки расположены в репозитории проекта [32].

7. Заключение

Сформирована открытая, многоуровневая, сегментированная, сервис–ориентированная архитектура программного комплекса, основанная на использовании технологий микросервисов, контейнерной виртуализации и обмена сообщениями по подписке. Космический сегмент включает программы–клиенты для управления бортовыми конечными устройствами, которые организованы в виде микросервисов на платформе контейнеризации Docker.

Программный комплекс УЕМКА реализует следующие функции:

- удаленный доступ к бортовым конечным устройствам (передача команд и данных между устройствами и наземными пунктами управления);
- управление устройствами с помощью единой централизованной консоли;
- автоматизированное управление бортовыми конечными устройствами при решении целевых задач (обработка изображений, параллельные вычисления и пр.);
- удаленная настройка конфигураций на пользовательских устройствах;
- подготовка, развертывание приложений и управление ими;
- выполнение пользовательских скриптов для автоматизации выполнения задач;
- удаленное устранение ошибок и отказов SMART-устройств;
- выборочное ограничение функциональных возможностей устройства;
- графический интерфейс пользователя наземного сегмента.

Представленный проект имеет открытый исходный код, что обеспечивает ряд преимуществ: программное обеспечение находится в общественном доступе, имеет свободное распространение и возможность модификации, все исходные коды доступны в виде публичного репозитория. Открытое программное ядро позволяет гибко и оперативно встраивать новые полезные нагрузки на борт спутника, экономя время разработчиков, а также дает возможность использовать современные программные методы и технологии, включая нейросетевые.

Результаты проекта программного комплекса УЕМКА используются в научно-исследовательских работах, курсовых проектах и выпускных квалификационных работах студентов кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана. Они также внедрены в учебный процесс по трем дисциплинам:

- применение открытого программного обеспечения для расчета конструкций РКТ;
- системный анализ изделий ракетно-космической техники;
- системное моделирование изделий ракетно-космической техники.

Прикладное использование проекта сосредоточено во взаимодействии с разработчиками космического электронного оборудования и обучающимися вузов по направлениям ракетно-космической отрасли.

Результаты работы представлены в сети Интернет в публичном репозитории GitLab [33] и, в рамках импортозамещения, зеркало в российском сервере хранения исходного кода GitFlic [34]. Проект активно развивается с привлечением контрибьюторов и специалистов из космической отрасли.

Список литературы / References

- [1]. Платформа LM50 для технологии SmartSat. [Электронный ресурс]: <https://www.lockheedmartin.com/en-us/products/satellite.html> (дата обращения 22.02.2024).
- [2]. More Capable LM 400 Satellite Bus Designed to Meet Urgent Mission Needs [Электронный ресурс]: <https://www.lockheedmartin.com/en-us/products/satellite.html> (дата обращения 24.12.2023).
- [3]. Lockheed Martin LINUSS Small Satellites Ready for 2021 Launch [Электронный ресурс]: <https://news.lockheedmartin.com/linuss-small-sats-mission> (дата обращения 10.04.2024).
- [4]. Борисов сообщил о планах создать в 2026 году завод по серийному производству спутников [Электронный ресурс]: <https://tass.ru/kosmos/17889029> (дата обращения 22.02.2024).
- [5]. «Don't try this at home» pilot for a Cognitive Cloud Computing in Space infrastructure IAC-22-B1.4 //73rd International Astronautical Congress (IAC 2022). – 2022.
- [6]. Edge computing in space: a machine learning approach for anomaly detection IAC-22-B5.1.6 //73rd International Astronautical Congress (IAC 2022). – 2022.
- [7]. Haiyang Chua, Xiaoyu Heb, Hongjiang Songc, Shaohua Baid. The Design of the Spacecraft Test System 4000 Based on Microservices Running in Cloud Environment IAC-22-B6.1.1.x68506 //73rd International Astronautical Congress (IAC 2022). – 2022.
- [8]. Проект F⁷ Flight Software & Embedded Systems Framework от НАСА и JPL. [Электронный ресурс]: <https://nasa.github.io/fprime/> (дата обращения 10.04.2024).
- [9]. QNX – система реального времени. [Электронный ресурс]: <https://blackberry.qnx.com/en> (дата обращения 10.04.2024).
- [10]. COSMOS API. [Электронный ресурс]: <https://ballaerospace.github.io/cosmos-website/docs/v4/json-api> (дата обращения 10.04.2024).
- [11]. XML Telemetric and Command Exchange. [Электронный ресурс]: <https://public.ccsds.org/Pubs/660x2g2.pdf> (дата обращения 10.04.2024).
- [12]. David J. Barnhart, David M. Shoemaker, Ellis T. King, Thomas “TJ” Logue, Michael J Lavis. LM LINUSS™ - Lockheed Martin In-space Upgrade Servicing System. 37th Annual Small Satellite Conference. [Электронный ресурс]: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=5648&context=smallsat> (дата обращения 22.03.2024).
- [13]. Core Flight System (cFS) [Электронный ресурс]: <https://cfs.gsfc.nasa.gov/> (дата обращения 12.01.2024).
- [14]. ПО для автопилотов PX4 [Электронный ресурс]: <https://github.com/PX4/PX4-Autopilot> (дата обращения 15.02.2024).
- [15]. ПО для автопилотов franca [Электронный ресурс]: <https://github.com/franca/franca/> (дата обращения 17.02.2024).
- [16]. Core Flight System (CFS) Command and Data Dictionary (CCDD) utility [Электронный ресурс]: <https://github.com/nasa/CCDD> (дата обращения 12.02.2024).
- [17]. Danda B. Rawat, Joel J.P.C. Rodrigues, Ivan Stojmenovic Cyber-Physical Systems From Theory to Practice. CRC Press 2016 579 p.
- [18]. Kiruthika Devaraj, Matt Ligon, Eric Blossom, Joseph Breu, Bryan Klofas, Kyle Colton, Ryan Kingsbury. Planet High Speed Radio: Crossing Gbps from a 3U Cubesat. 33rd Annual AIAA/USU Conference on Small Satellites. [Электронный ресурс]: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=4405&context=smallsat> (дата обращения 22.03.2024).
- [19]. Satellite Eutelsat 10B [Электронный ресурс]: https://www.eutelsat.com/files/PDF/brochures/EUTELSAT_SATELLITE_E10B.pdf (дата обращения 22.03.2024).

- [20]. Жданова К.А., Щеглов Г.А. Разработка вычислительного модуля для малого космического аппарата класса CubeSat // XLV Академические чтения по космонавтике (Королёвские чтения — 2022): сб. тез.: в 4 т. М.: Изд-во МГТУ им. Н.Э. Баумана, 2022. Т. 3. С. 241–243.
- [21]. Ссылка на README.md файл репозитория проекта UEMKA [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/blob/devel/README.md (дата обращения 29.03.2024).
- [22]. Парминдер Сингх Кочер Микросервисы и контейнеры Docker ДМК Пресс 2019 240 с.
- [23]. Крис Ричардсон. Микросервисы. Паттерны разработки и рефакторинга Питер, 2022 544 с. Подробнее: [Электронный ресурс]: <https://www.labirint.ru/books/707677/> (дата обращения 12.03.2024).
- [24]. SocketCAN userspace utilities and tools [Электронный ресурс]: <https://github.com/linux-can/can-utils> (дата обращения 14.01.2024).
- [25]. Дисс. на соискание уч. ст. к.т.н. «Методика проектирования наноспутника с солнечной энергодвигательной установкой» [Электронный ресурс]: https://mai.ru/upload/iblock/183/vpbzeo51l25g7p8a6bq03ml3hm0sj3n7/Dissertatsiya_ZHumaev.pdf (дата обращения 19.03.2024).
- [26]. Сервис численного моделирования КА [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/satellite_model?ref_type=heads (дата обращения 15.10.2024).
- [27]. Библиотеки для работы численной модели [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/libs (дата обращения 29.03.2024).
- [28]. Сервис с нейросетью распознавания изображений. Устройство neural service [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/neural_service (дата обращения 10.10.2024).
- [29]. M. Bernou, A. Ampatzoglou, S. Vellas, K. Panopoulou, G. Lentaris, D. Soudris Spacecraft as a service, an open-source approach // 29th IAC Symposium On Small Satellite Missions (B4) IAC-22,B4,IP,4,x69307 URL: <https://iafastro.directory/iac/paper/id/69307/abstract-pdf/IAC-22,B4,IP,4,x69307.brief.pdf?2022-04-05.11:15:32> (дата обращения 15.03.2024).
- [30]. Каталог с приложением интерфейса PULT [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/pult (дата обращения 10.10.2024).
- [31]. Графический интерфейс управления CAN-шиной [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/GUI_rules_CAN (дата обращения 10.10.2024).
- [32]. Примеры вариантов использования кода проекта UEMKA [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/docs/usage_examples (дата обращения 29.03.2024).
- [33]. Ссылка в сети Интернет на публикацию открытой библиотеки в публичном репозитории GitLab [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems (дата обращения 18.02.2024).
- [34]. Ссылка в сети Интернет на публикацию открытой библиотеки в российском сервере хранения исходного кода GitFlic: [Электронный ресурс]: <https://gitflic.ru/project/zhumaev/acs?branch=devel> (дата обращения 19.02.2024).

Информация об авторах / Information about authors

Георгий Александрович ЩЕГЛОВ – доктор технических наук, профессор, профессор кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана с 2012 года. Сфера научных интересов: проектирование аэрокосмических систем, системы автоматизированного проектирования и расчета элементов конструкций аэрокосмических систем.

Georgy Alexandrovich SHCHEGLOV – Dr. Sci. (Tech.), Professor, Professor of the Department of Aerospace Systems at Bauman Moscow State Technical University since 2012. Research interests: design of aerospace systems, computer-aided design systems and calculation of structural elements of aerospace systems.

Кристина Александровна ЖДАНОВА – инженер, аспирант кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана. Сфера научных интересов: орбитальные высокопроизводительные вычисления, системное проектирование и инженерия, встраиваемые системы, CubeSat, проектирование аэрокосмических систем.

Kristina Alexandrovna ZHDANOVA is an engineer, postgraduate student of the Aerospace Systems Department of Bauman Moscow State Technical University. Research interests: embedded systems, CubeSat, design of aerospace systems.

Зайнулла Серикович ЖУМАЕВ – кандидат технических наук, преподаватель кафедры «Аэрокосмические системы» МГТУ им. Н. Э. Баумана. Сфера научных интересов: системное проектирование, математическое моделирование малых космических аппаратов, разработка алгоритмов управления, открытое программное обеспечение.

Zaynulla Serikovich ZHUMAEV – Cand. Sci. (Tech.), Lecturer of the Aerospace Systems Department, Bauman Moscow State Technical University. Research interests: system design, mathematical modeling of small spacecraft, development of control algorithms, open source software.

Никита Дмитриевич КАМЕНЕВ – аспирант кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана. Сфера научных интересов: нейронная сеть, искусственный интеллект, проектирование аэрокосмических систем.

Nikita Dmitrievich KAMENEV is a postgraduate student of the Aerospace Systems Department of Bauman Moscow State Technical University. Research interests: neural network, artificial intelligence, design of aerospace systems.