

УДК 681.5

<https://doi.org/10.21869/2223-1560-2024-28-2-92-113>

Оценка уязвимости протоколов обмена данными БПЛА для обеспечения повышения стабильности работы в нагруженных режимах

С. Г. Черный^{1,3} ✉, Т.В. Зонтова², Н. В. Шапаренко³

¹ Керченский государственный морской технологический университет,
ул. Орджоникидзе, д. 82, г. Керчь 298309, Республика Крым, Российская Федерация

² Черноморское высшее военно-морское училище имени П.С. Нахимова,
ул. Дыбенко, д. 1а, г. Севастополь 299028, Российская Федерация

³ Государственный университет морского и речного флота имени адмирала С.О. Макарова,
ул. Двинская, д. 5/7, г. Санкт-Петербург 198035, Российская Федерация

✉ e-mail: sergiiblack@gmail.com

Резюме

Цель работы. В настоящее время активно растёт интерес к использованию дронов в различных сферах. Причины связаны с непрерывным ростом технологий, с появлением быстрых микропроцессоров, которые обеспечивают автономное управление несколькими системами связи. Мониторинг, строительство, контроль и слежение – лишь некоторые из областей, в которых использование БПЛА становится обычным явлением. Цель работы заключается в изучении работы нагруженности протоколов обмена информации на различных уровнях взаимодействия и предложить варианты повышения надежности взаимодействия на основе гибридных технологий. Задачами исследования является детализация протоколов обмена данными в БПЛА на различных уровнях, с учетом анализа структуры передаваемой информации и гибридной модели.

Методы. В статье рассматриваются протоколы, задействованные в работе БПЛА на разных уровнях, их особенности, преимущества и недостатки, а также их свои восстановление связи. Использование реалистичных технологических особенностей беспилотных летательных аппаратов для тестирования моделей и методов может оказаться весьма актуальным для практических целей в различных отраслях от гражданских до военных.

Результаты. В статье представлена детализация протоколов связи и обмена данными, которые используются в системах БПЛА, а также их производительность.

Заключение. Для качественной оценки влияния на безопасность информации, предлагается ввести гибридные модели компоненты, которые динамически адаптируют протоколы обмена данными на основе анализа угроз в режиме работы в реальном времени и возможностей системы. Рациональное использование энергии имеет решающее значение для осуществления эффективного и безопасного процесса работы БПЛА. Затронуты и рассмотрены вопросы защищенности канала передачи загруженности протокола с позиции затрат энергетических ресурсов.

Ключевые слова: БПЛА; протокол; уязвимость; реальное время; угроза; информационная безопасность; загруженность протоколов.

Конфликт интересов: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Черный С. Г., Зонтова Т.В., Шапаренко Н. В. Оценка уязвимости протоколов обмена данными БПЛА для обеспечения повышения стабильности работы в нагруженных режимах // Известия Юго-Западного государственного университета. 2024. Т. 28, №2. С. 92-113. <https://doi.org/10.21869/2223-1560-2024-28-2-92-113>.

Поступила в редакцию 17.04.2024

Подписана в печать 27.05.2024

Опубликована 25.06.2024

Vulnerability assessment of UAV data exchange protocols to ensure increased stability of operation in loaded modes

Sergey G. Cherny ^{1,3} ✉, Tatyana V. Zontova ², Nikita V. Shaparenko ³

¹ Kerch State Marine Technological University
82 Ordzhonikidze str., Kerch 298309, Republic of Crimea, Russian Federation

² Black Sea Higher Naval School named after P.S. Nakhimov,
1a, Dybenko str., Sevastopol 299028, Russian Federation

³ State University of Marine and River Fleet named after admiral S.O. Makarov
5/7 Dvinskaya str., St. Petersburg 198035, Russian Federation

✉ e-mail: sergiiblack@gmail.com

Abstract

Purpose of research. Currently, interest in the use of drones in various fields is actively growing. The reasons are related to the continuous growth of technology, with the advent of fast microprocessors that provide autonomous control of multiple communication systems. Monitoring, construction, control and surveillance are just some of the areas in which the use of UAVs is becoming commonplace. The purpose of the work is to study the work of loaded information exchange protocols at various levels of interaction and to propose options for increasing the reliability of interaction based on hybrid technologies. The article provides details of the communication and data exchange protocols used in UAV systems, as well as their performance.

Methods. The article discusses the protocols involved in the operation of UAVs at different levels, their features, advantages and disadvantages, as well as their failures to restore communication. Using realistic technological features of unmanned aerial vehicles to test models and methods can be very relevant for practical purposes in various industries from civil to military.

Results. The objectives of the study are to detail the data exchange protocols in UAVs at various levels, taking into account the analysis of the structure of the transmitted information and the hybrid model.

Conclusion. To qualitatively assess the impact on information security, it is proposed to introduce hybrid models of components that dynamically adapt data exchange protocols based on real-time threat analysis and system capabilities. Efficient use of energy is critical to efficient and safe UAV operation. The issues of security of the transmission channel of the protocol load from the point of view of the cost of energy resources are touched upon and considered.

Keywords: UAV; protocol; vulnerability; real time; threat; information security; protocol load.

Conflict of interest. The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Cherny S. G., Zontova T. V., Shaparenko N. V. Vulnerability assessment of UAV data exchange protocols to ensure increased stability of operation in loaded modes. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2024; 28(2): 92-113 (In Russ.). <https://doi.org/10.21869/2223-1560-2024-28-2-92-113>.

Received 17.04.2024

Accepted 27.05.2024

Published 25.06.2024

Введение

В наше время беспилотные летательные аппараты (БПЛА) стали неотъемлемыми инструментами в самых различных областях, начиная от гражданских задач – визуальное наблюдение при помощи в случае ЧП, элементами транспортной логистики, заканчивая военной разведкой и участием в полноценных военных конфликтах [1-3].

В ходе реализации работы использованы самодельные основы БПЛА в качестве материнских плат на основе Raspberry Pi 3 и STMicroelectronics STM32F407 Discovery Board на базе ARM Cortex M4 процессора (STM32F407VGT6). В качестве оболочки для самодельного дрона использовался Ardupilot. Для анализа протоколов передачи информации низкого (логического) уровня использовался осциллограф Hantek DSO-6022BE и логический анализатор ChipDipDac RDC2-0064. При работе с высокими скоростями передачи данных или большим объемом данных возникают некоторые проблемы с записью и декодированием в реальном времени, декодирование зашифрованных каналов связи также занимает время и вычислительные ресурсы, поэтому в своей рабо-

те будет упор на работу логического анализатора.

1. Отображение записанных выборок:

- захваченные данные отображаются на экране в виде временных диаграмм или графиков;

- во время отображения данных анализатор может обнаруживать и декодировать протоколы передачи данных, такие как UART, SPI, I2C, CAN и другие. Это позволяет пользователю анализировать передаваемые данные и последовательность действий между устройствами.

2. Анализ выборок в специализированном ПО (PulseView):

- PulseView – программное обеспечение с открытым исходным кодом для анализа данных, полученных от логических анализаторов. Оно позволяет визуализировать данные, анализировать протоколы и искать аномалии в сигналах;

- непосредственно в PulseView можно настраивать параметры анализа, фильтровать данные, синхронизировать различные каналы сигналов и производить другие операции для более детального анализа данных;

- можно визуально анализировать захваченный трафик, проверять целостность информации, выводить содержи-

мое сообщений захваченных данных, декодировать сообщения протоколов, используя фильтры.

Материалы и методы

Протоколы обмена данными на физическом уровне в БПЛА

В БПЛА различных производителей зачастую используются следующие основные протоколы на логическом уровне:

SPI (Serial Peripheral Interface): синхронный последовательный интерфейс, который был разработан для обмена данными между микроконтроллерами и периферийными устройствами, такими как сенсоры, датчики, дисплеи и другие. Он работает в полудуплексном режиме.

Архитектурно, SPI устроен так, что использует 4 сигнальные линии:

- SCLK (синхронный тактовый сигнал);
- MOSI (Master Out Slave In);
- MISO (Master In Slave Out);
- SS/CS (Slave Select \ Chip Select).

Сообщение в протоколе SPI представляет собой последовательность бит, которые передаются по линиям MOSI (Master Out Slave In) и MISO (Master In Slave Out).

Процесс отправления сообщения длиной 8-бит, которое передавалось по положительному фронту SCLK, а данные считывались по спадающему фронту SCLK [1, 4-7]. Главный процессор (master) отправляет байт данных 0xAB (10101011 в двоичной системе) на ли-

нию MOSI, а датчик или подчинённый процессор (slave) отвечает этим же байтом на линии MISO – процесс называется «эхо» и может использоваться при проверке работоспособности линии. В таком случае каждый бит данных передается на каждом тактовом переходе. Сначала master отправляет старший бит (MSB – Most Significant Bit), а slave считывает его. Затем передается следующий бит и так далее, пока не переданы все 8 бит сообщения. Необходимо согласовать настройки передачи данных между slave и master, полярность сигнала SCLK, так как в ином случае правильное чтение и запись информации могут быть нарушены.

I²C (Inter-Integrated Circuit): является синхронным последовательным интерфейсом, который используется для связи между микроконтроллерами и различными периферийными устройствами, такими как датчики, память, дисплеи, а также для обмена данными между несколькими микроконтроллерами.

Немаловажной особенностью для использования в БПЛА является энергосбережение в работе протокола датчиками. Сообщение в протоколе *I²C* (Inter-Integrated Circuit) представляет собой последовательность бит, передаваемых последовательно по двум линиям - линия данных (SDA) и линия тактирования сигнала (SCL). Протокол *I²C* работает в синхронном режиме, где тактовый сигнал SCL используется для синхронизации передачи данных между устройствами. *I²C* шина работает следующим образом.

1. Ведущие и ведомые устройства:

В шине I²C есть два типа устройств – ведущие (master) и ведомые (slave).

Master – устройство, которое иницирует коммуникацию и контролирует передачу данных. Slave – устройства, которые отвечают на запросы мастера и передают данные. Логично, что все датчики в БПЛА будут ведомыми.

2. Две линии: Шина I2C состоит из двух линий - линия данных (SDA) и линия тактирования (SCL). Линия данных (SDA) используется для передачи данных, а линия тактирования (SCL) используется для синхронизации передачи данных между устройствами.

3. Состояние «старт-стоп»: Передача данных в шине I²C осуществляется по принципу смены состояний «старт-стоп».

Прежде чем начать передачу данных, ведущий отправляет условие "старт" (START) на шину, чтобы предупредить все устройства о начале передачи. После завершения передачи данных ведущий отправляет условие "стоп" (STOP) на шину, чтобы завершить передачу.

4. 7-битные и 10-битные адреса: Каждое устройство на шине I²C имеет свой уникальный адрес, по которому мастер может обратиться к нему. 7-битные адреса наиболее распространены, но также существуют 10-битные адреса для большего количества подключаемых устройств.

5. ACK/NACK: после передачи каждого байта данных, получающее ведомое устройство обязано отправить биты

подтверждения ACK (Acknowledgment) или отказ NACK (Not-Acknowledgment) ведущему. ACK символизирует ведущему, что передача прошла успешно, а NACK указывает на ошибку или отсутствие устройства.

6. Биты данных и тактовый сигнал: В I²C данные передаются последовательно, от старших битов к младшим, при этом передача данных на линии SDA может изменяться только, когда SCL имеет низкий логический уровень.

7. Скорости передачи: Шина I²C поддерживает различные скорости передачи данных, обычно от нескольких килобит в секунду до нескольких мегабит в секунду. «Узким горлышком» в передаче выступает производительность ведомых устройств – датчиков.

Сообщение в I²C состоит из следующих элементов (табл.1) [1, 7-9]:

1. Начальное условие (Start Condition): Ведущий (master) генерирует биты инициализации на линии SDA, чтобы указать на начало передачи данных. Стартовое условие состоит из перехода с высокого уровня (логической 1) на низкий уровень (логический 0) на линии SDA, при удержании линии SCL в высоком уровне.

2. Адрес устройства (Address Byte): после стартового условия master сообщает адрес устройства, с которым он хочет общаться, он состоит из 7 бит, и младший значимый бит инструкции указывает, является ли запрашиваемая операция - чтением или записью. Если младший бит адреса равен 0, это озна-

чает операцию записи, если равен 1 – операцию чтения.

3. *Бит подтверждения (Acknowledge Bit)*: после передачи адреса устройства, получатель отправляет бит подтверждения, чтобы указать, что он готов принять данные. Если устройство готово принять данные, оно устанавливает линию SDA в низкий уровень (логический 0) во время следующего тактового импульса. Если устройство не готово принять данные или произошла ошибка, линия SDA остается в высоком уровне (логическая 1).

4. *Данные (Data Bytes)*: после передачи адреса и получения бита подтверждения, master и устройство могут обмениваться данными. Каждый байт данных передается последовательно, начиная со старшего бита (MSB) и заканчивая младшим значимым битом (LSB – Least Significant Bit).

5. *Бит подтверждения (Acknowledge Bit)*: после передачи каждого байта данных, получатель также отправляет бит подтверждения, чтобы указать, что он успешно принял данные.

6. *Конечное условие (Stop Condition)*: Master генерирует сигнал остановки, чтобы указать на окончание передачи данных. Конечное условие состоит из перехода с низкого уровня (логический 0) на высокий уровень (логический 1) на линии SDA, при удержании линии SCL в высоком уровне.

I²C поддерживает много ведущих устройств на линии одновременно, однако в БПЛА стоит придерживаться одного master на линии, чтобы не приводить к конфликтам и коллизиям на шине, или проводить грамотное планирование и управление шиной во избежание ошибок и неправильного чтения \ записи данных.

Таблица 1. Структура передачи одного байта данных (0xAB) ведущим устройством клиенту

Table 1. Structure of transmission of one data byte (0xAB) from the master to the client

Начало / Start	Адрес / Address	Ack (бит под- тверждения) / Ack (confirma- tion bit)	Данные (0xAB) / Data (0xAB)	Ack (бит под- тверждения) / Ack (confirma- tion bit)	Конец / Stop
S	1010100	0	101010111	0	P

Примечание: **S** – Начальное условие; **P** – Конечное условие; **Ack** – Бит подтверждения.

UART (Universal Asynchronous Receiver/Transmitter) – асинхронный последовательный интерфейс, который используется для передачи данных между устройствами. Он широко применяется в связке с компьютерами, периферий-

ными устройствами и микроконтроллерами, работая в полудуплексном режиме, он использует 2 сигнальные линии – Tx (сигнал передачи) и Rx (сигнал приёма) (таб.2). Сообщение в UART состоит из следующих элементов:

1. *Начальный бит (Start Bit)*: всегда имеет значение 0. Этот бит служит для синхронизации приемника и указывает начало передачи символа.

2. *Данные (Data Bits)*: Количество данных битов зависит от настроек UART и может быть 5, 6, 7 или 8 бит. В БПЛА часто используется 7 и 8 бит длины сообщений.

3. *Бит четности (Parity Bit)*: это опциональный бит, который может быть использован для обеспечения контроля четности данных. В зависимости от выбранного режима (четный или нечетный), бит четности может быть установлен как 0 или 1.

4. *Конечный бит (Stop Bit)*: обычно имеет значение 1 и используется для завершения передачи символа.

Таблица 2. Структура сообщения UART с 8-битными данными, без бита четности и одним конечным битом

Table 2. UART message structure with 8-bit data, no parity bit and one tail bit

Start Bit	Data Bits (8 бит)	Stop bit
0	01101011	1

В данном сообщении содержится 8 бит данных (01101011), стартовый бит (0) и один стоповый бит (1).

Каждый символ передается последовательно, и после передачи одного символа следует небольшая пауза (интервал между символами), прежде чем начнется передача следующего символа.

1. Для скорости 9600 бит/с (9600 baud) длительность паузы составила около 110 микросекунд.

2. Для скорости 115200 бит/с (115200 baud) длительность паузы составила около 9 микросекунд.

Структура сообщения в UART может изменяться в зависимости от настроек протокола и устройств, между которыми происходит обмен данными. При использовании UART нужно обязательно согласовывать настройки (битовую скорость, количество данных бит и т.д.) между отправителем и получателем для успешной передачи и приема данных.

лем для успешной передачи и приема данных.

Jitter (фазовое дрожание цифрового сигнала данных) — непредсказуемое изменение временных интервалов между символами в передаче данных через UART и/или другой протокол, которое может возникать из-за неточности внутренних часов устройств, электромагнитных помех, а также шумов на линиях связи и других факторов. Джиттер может приводить к ошибкам в приеме данных и вызывать некорректную интерпретацию символов (рис.1) [5-7].

Логический анализатор при скорости в 115200 бод не увидел границы сообщений – старт и стоп биты. Поэтому, показывает предупреждения Frame Error, которые символизируют искажения полученной информации и требуют обратить внимание на этот фрагмент данных. Если отобразить визуально – то

полученные данные при скорости в 9600 бод, декодированные как полученные при 115200 бод, будут отображаться совсем иначе.

Джиттер может также вызывать искажения Frame (пакета), он зависит от точности и стабильности работы системы. В БПЛА же, зачастую, посредством электромагнитных помех и воздействий температуры могут создаваться потери и искажения сообщений. Используя специальные микросхемы, которые обеспечивают стабильность часов (точное внешнее тактирование с кварцевого резонатора или генератора), выполнять коррекцию уже существующих ошибок (ЕСС) возможно [10-15].

Говоря про разные платформы для производства БПЛА, стоит упомянуть, что в них может быть различное количество интерфейсов – в зависимости от центрального процессора, сопроцессоров или другой встроенной логики и периферии. Применимо к Raspberry Pi 3 и дрону, основанном на микроконтроллере STM32F407VGT6, можно составить сравнительную таблицу (табл. 3).

Raspberry Pi 3 имеет только один встроенный UART порт (UART0), в то время как STM32F407VGT6 обладает несколькими встроенными UART портами. STM32F407VGT6 предоставляет больше гибкости в выборе количества бит данных (8 или 9) и стоп-битов (1, 1.5 или 2).

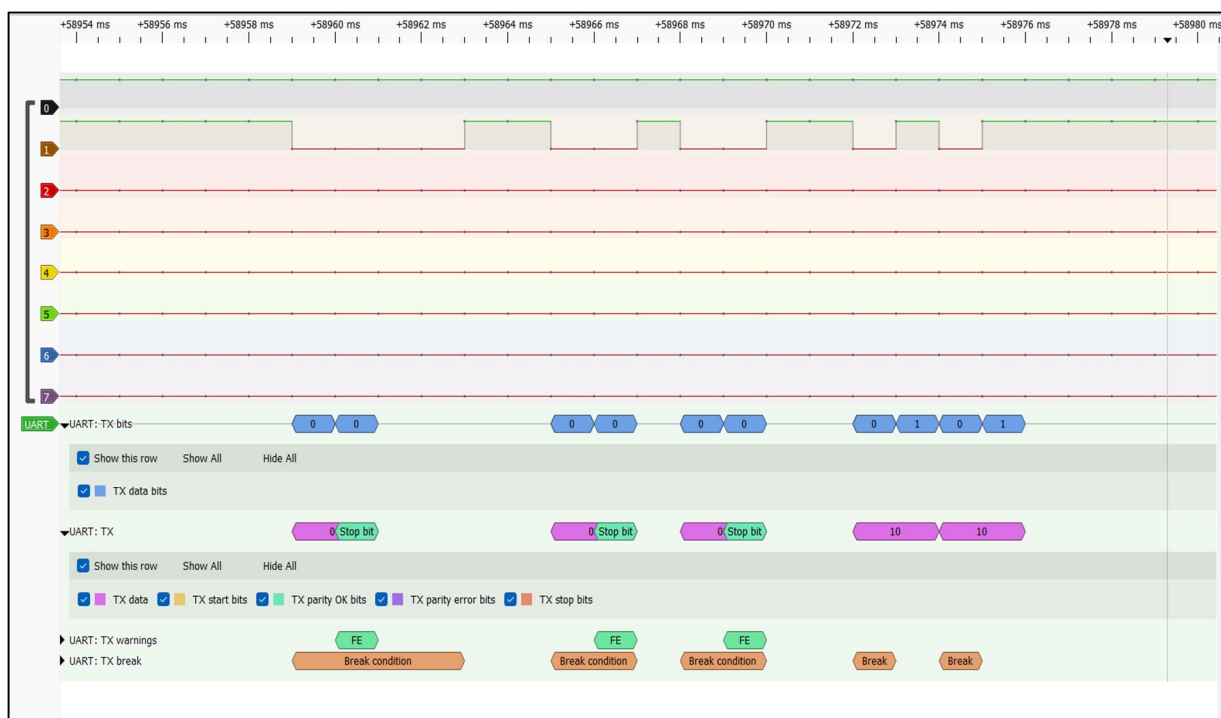


Рис. 1. Снимок выборки, декодированной как UART на скорости 115200 бод

Fig. 1. Snapshot of sample decoded as UART at 115200 baud

Таблица 3. Сравнительная характеристика поддержки UART**Table 3.** Comparative characteristics of UART support

Характеристика / Characteristic	Raspberry Pi 3	STM32F415VGT6
Количество UART	1 встроенный UART (UART0)	Несколько встроенных UART портов
Скорость передачи данных	Различные скорости до 921600 bps	Различные скорости до нескольких мегабит в секунду
Биты данных	8	8 или 9 (по выбору)
Биты контроля четности	Отсутствует	Можно выбрать или отключить
Стоп-биты	1 или 2	1, 1.5 или 2 (по выбору)

Джиттер также может вызывать смещение временных интервалов в выборке логического анализатора между символами, что приводит к неправильному распознаванию символов. Могут искажённо интерпретироваться стартовый бит, биты данных и стоп-бит, что обязательно приведёт к ошибкам в декодировании передаваемой информации. В случае, если джиттер слишком велик, логический анализатор может потерять символы, что приведет к потере части или всего сообщения.

Потеря данных в джиттере особенно опасна в случае передачи критически важной информации, такой как управляющие команды или показания датчиков, а это, в свою очередь, может привести к неправильным действиям или потере ценных данных для управления БПЛА.

Джиттер может сильно ухудшить надежность связи по UART между БПЛА и его контроллером или другими устройствами, а это может привести к

потере связи в критических моментах, что может вызвать сбой в работе БПЛА или даже его потерю.

Результаты и их обсуждения

Ошибки интерпретирования стартового и стоп-битов приведут к неправильному определению границы символов и протокольных пакетов, а это, в свою очередь, помешает декодированию всего сообщения [1,4,16-18].

Ardupilot — открытое программное обеспечение (распространяемое по лицензии «Open Source» – открытые (бесплатные и опубликованные) исходные коды для некоммерческого использования) для автономного управления БПЛА и другими робототехническими платформами.

Чтобы собрать полноценный БПЛА на основе Ardupilot требуется:

– Основная плата с программным комплексом Ardupilot. Для тестирования собран аналог Pixhawk 4.

– Контроллер полёта (плата-пульт, которая позволяет управлять перемещением самодельного БПЛА – это может быть Pixhawk, Pixhawk 2, Pixhawk 4 и другие. Пульт пересылает данные об управлении на дрон как Mavlink сообщения) (рис.2).

Информационные протоколы

MAVLink (Micro Air Vehicle Communication Protocol) – это протокол обмена сообщениями для беспилотных летательных аппаратов, который разрабатывался для обеспечения связи между автономными системами и внешними устройствами [7-9].

MavLink разрабатывался с учетом ограниченных ресурсов микроконтроллеров, которые используются в автономных системах, что, несомненно, полезно в работе с БПЛА.

Mavlink – информационный протокол, который основывается на более низкоуровневых протоколах. Поэтому, Mavlink может быть реализован на базе UART, SPI, I2C, в зависимости от возможностей, процессорной базы и архитектуры дрона.

Например, для обмена данными между контроллером полета и компьютером на земле часто используется UART, а для взаимодействия с датчиками могут применяться различные интерфейсы, такие как I²C или SPI.

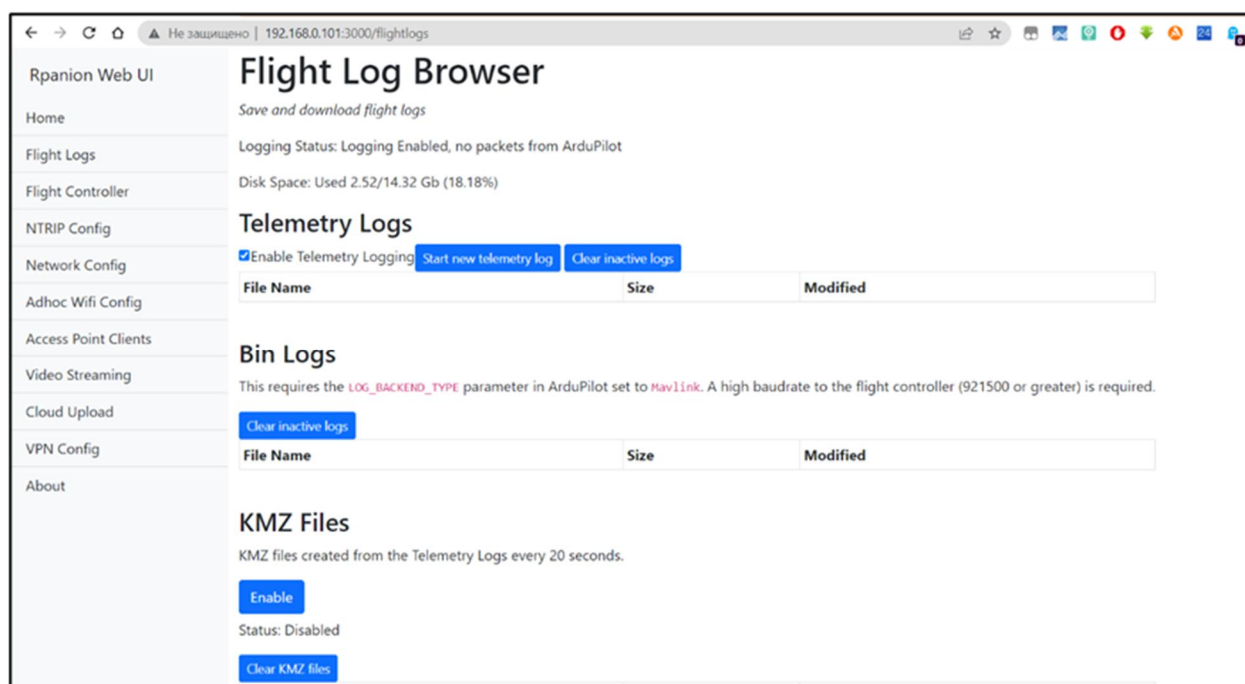


Рис. 2. Интерфейс Ardupilot

Fig. 2. Ardupilot interface

MavLink является важным протоколом для обеспечения связи между

беспилотными летательными аппаратами и внешними устройствами, позволяя

передавать и получать данные о состоянии, навигации и управлении, что делает его основой для разработки автономных и управляемых БПЛА.

Разберу отправленное сообщение «HEARTBEAT»:

Оно отправляется периодически системой БПЛА и служит для информирования внешнего устройства (например, наземной станции), о том что дрон работает и сообщает базовую информацию о себе, нужно это, чтобы контролировать связь и статус БПЛА в реальном времени и принимать соответствующие действия на основе этих данных.

MQTT (Message Queuing Telemetry Transport)

MQTT (Message Queuing Telemetry Transport) — это протокол обмена сообщениями, который был разработан для передачи данных между устройствами в условиях ограниченной пропускной способности или ненадежным соединением.

Протокол MQTT работает по алгоритму:

1. *Подписка ведущим устройством на шину ведомого (Subscribe):*

Клиент может подписаться на определенную шину, чтобы получать информацию или команды от брокера сообщений. Команда подписки содержит имя шины и уровень желаемого QoS (Quality of Service) – уровня обслуживания, который определяет гарантии доставки сообщений. Чтобы подписаться, необходимо отправить команду *SUBSCRIBE*.

2. *Опубликовать сообщение (Publish):*

Клиент может отправить сообщение на определённую шину, чтобы передать информацию или команду другим клиентам, которые подписаны на эту шину.

Опубликованное сообщение содержит имя шины, данные и уровень QoS. Чтобы отправить сообщения, необходимо отправить команду *PUBLISH*.

3. *Отписаться от топика (Unsubscribe):*

Клиент может отписаться от определенной шины, чтобы больше не получать сообщения с этой шины и от брокера сообщений. Отписка осуществляется с помощью команды *UNSUBSCRIBE*.

4. *Получение подтверждения (Acknowledgement).*

В данном сообщении, наземная станция запрашивает у БПЛА текущие инструкции. Клиент публикует сообщение с запросом на шину "commands":

PUBLISH

Topic: commands

Data: "Request instructions"

QoS: 1

БПЛА через брокера сообщений получает это сообщение и высылает в ответ выполняемые инструкции на шину Instructions (взлёт, полёт на координаты Санкт-Петербурга, центр города, посадка):

PUBLISH

Topic: instructions

Data: "Take off, fly to coordinates (59.9343, 30.3351), land"

QoS: 1

Синтаксис команд и названия топиков зависят от версии управляющего ПО, платы контроллера дрона и т.д., важно чтобы они были синхронизированы. Сам протокол MQTT может использоваться поверх различных физических интерфейсов, таких как UART, SPI и Ethernet.

MQTT, будучи разработанным для работы в мире IoT («Internet of Things» с англ. – Интернета вещей) – более гибкое архитектурное решение, он может использоваться в системах реального времени (операционных системах реального времени). Также он более энергосберегающий и адаптирован под централизованные системы с ведущими и ведомыми устройствами, где брокеры

сообщений связывают всех воедино, что позволяет применять его гибридно.

MavLink же был разработан для БПЛА, его архитектура и структура сообщений оптимизированы для использования в ограниченных ресурсах микроконтроллеров, что делает его предпочтительным для автономных систем с высокой задержкой и пропускной способностью.

TCP\UDP (табл. 4)

TCP (Transmission Control Protocol) и UDP (User Datagram Protocol) – это 2 основных протокола транспортного уровня, используемых в сетевых коммуникациях. В БПЛА они используются для обмена данными между устройствами и серверами или управляющими центрами \ наземными станциями [16-20].

Таблица 4. Сравнение транспортных протоколов TCP и UDP

Table 4. Comparison of TCP and UDP transport protocols

Протокол / Characteristic	Надежность / Protocol Reliability	Установка соединения / Connection setup	Потокоориенти- рованность / Flow orientation	Тип данных / Data type
TCP (Transmission Control Protocol)	Гарантирует надежную передачу данных. Переотправляет потерянные или поврежденные пакеты	Требуется установка соединения перед передачей данных, что вызывает некоторую задержку	Потокоориентированный: передает данные в виде непрерывного потока	Поток данных (поток байтов)
UDP (User Datagram Protocol)	Не гарантирует доставку данных и их упорядочивание	Не требует установки соединения, что делает его быстрее и более эффективным	Не потокоориентированный: работает с дейтаграммами, которые могут быть обработаны независимо	Дейтаграммы (небольшие пакеты данных)

В БПЛА логично использовать TCP соединения в случае, когда надёжность передачи данных является критическим аспектом и недопустима потеря данных. Например, команды управления БПЛА должны отправляться в TCP, чтобы все пакеты в запросе были точно доставлены и выполнены в нужном порядке.

UDP же позволяет в условиях низкой задержки времени передачи данных передавать данные в реальном времени – это важно для передачи телеметрических данных о полёте или видео с камеры в БПЛА, где высокая скорость и минимальная задержка более критична, чем надёжность передачи информации.

В зависимости от конкретного сценария использования, можно комбинировать оба протокола: TCP для надёжной передачи важных данных и UDP для передачи данных с низкой задержкой в реальном времени.

Но помимо ручного комбинирования протоколов, есть и «легковесные реализации» для IP протокола целиком.

LWIP (Lightweight IP) – стек протоколов для работы с сетевыми протоколами, который широко используется во встроенных системах и микроконтроллерах, включая применение в БПЛА.

Как и протокол IP, по умолчанию LWIP не предоставляет механизмов безопасности, таких как шифрование данных или аутентификация. Это означает, что безопасность LWIP зависит от других слоев стека протоколов, таких как TLS \ SSL для обеспечения безопасности данных при использовании TCP. Администратору необходимо обеспечить правильную конфигурацию и реализацию служб безопасности в БПЛА с использованием LWIP, чтобы защитить данные от несанкционированного доступа и перехвата самостоятельно

Структура пакета IP (и LWIP)

IP-пакет состоит из заголовка и данных. IP-пакет имеет следующую структуру:



Рис. 3. Структура пакета IP (и LWIP)

Fig. 3. IP (and LWIP) Packet Structure

Он состоит из основных элементов:

1. Версия и длина: определяет версию протокола IP и длину заголовка пакета.

2. Тип сервиса (обслуживания): используется для указания приоритета обработки пакета в сети.

3. Длина пакета: указывает общую длину пакета (в байтах), включая заголовок и данные.

4. Идентификатор пакета: идентификатор, присвоенный пакету для его идентификации.

5. Флаги и смещение: используются при фрагментации больших пакетов.

6. Время жизни (TTL): указывает количество шагов, которые пакет может сделать по сети, прежде чем будет отброшен.

7. Протокол IP: определяет протокол верхнего уровня, к которому передаются данные из поля "Данные".

8. Контрольная сумма: используется для проверки целостности заголовка пакета.

9. IP-адрес отправителя и получателя: указываются IP-адреса отправителя и получателя пакета.

10. Дополнительные опции: могут присутствовать в заголовке и содержат дополнительные настройки или параметры.

11. Данные: содержимое пакета, которое будет передано протоколу верхнего уровня (например, TCP или UDP).

В LWIP соблюдается формат пакетов, как и теоретические постулаты модели OSI, он оптимизирован для работы на устройствах с ограниченными ресурсами, но в отличие от полноценного сте-

ка IP, LWIP не поддерживает некоторые продвинутое и менее используемые функциональности, в частности [15-20]:

1. IP-фрагментацию

LWIP не поддерживает автоматическую фрагментацию и сборку IP-пакетов, что означает, что пакеты IP должны быть достаточно маленькими, чтобы поместиться во все промежуточные сети по пути следования до конечного получателя. В полноценном стеке IP есть механизмы автоматической фрагментации и сборки пакетов для обеспечения передачи больших данных.

2. IPsec (IP Security):

LWIP не включает поддержку IPSec, протокола, предназначенного для обеспечения безопасности передачи данных путем шифрования и аутентификации IP-пакетов. IPSec – это продвинутое расширение протокола IP, которое защищает данные от несанкционированного доступа.

3. ICMP (Internet Control Message Protocol):

LWIP поддерживает основные типы ICMP-сообщений, такие как эхо-запрос (echo-request/reply), но может не поддерживать все виды ICMP-сообщений, которые могут быть доступны в полноценном стеке IP.

4. Multicast и Broadcast:

LWIP может не поддерживать полноценную обработку широковещательных пакетов и адресацию группе абонентов.

5. IPv6 Extension Headers:

В LWIP нет повсеместной реализации протокола IPv6 не реализована

6. ARP (Address Resolution Protocol):

LWIP может поддерживать ARP только для основной функциональности, такой как разрешение MAC-адресов, но может быть ограничен в дополнительных возможностях, связанных с ARP.

LWIP не является заменой протокола IP (Internet Protocol). Вместо этого, LWIP позиционируется как отличная реализация стека протоколов TCP/IP, включая IP, для встраиваемых систем и микроконтроллеров с ограниченными ресурсами. Основное предназначение LWIP – обеспечить поддержку сетевых протоколов в ограниченных условиях, где полный стек TCP/IP слишком ресурсоемкий.

Негативные факторы ухудшения связи в БПЛА

Негативные факторы, которые могут ухудшить связь в БПЛА (Беспилот-

ные летательные аппараты), включают холодную пайку, дребезг контактов и помехи электромагнитного излучения (ПЭМИН). Холодная пайка – ошибка в процессе пайки, когда место пайки и контакт недостаточно прогревается или не образуется надлежащее соединение между элементами. Оно негативно влияет на передачу информации, в месте плохого контакта может искажаться проходимый сигнал (рис. 4).

Холодная пайка может привести к ненадежным соединениям и повышенному сопротивлению на паянных контактах, что может вызвать плохую передачу сигналов и ухудшение связи. Для борьбы с ней следует использовать качественные припои, флюсы, нагревать место соединений до правильной температуры и обеспечивать надлежащую связь между паяными контактами, прибегать к автоматизации процесса пайки.

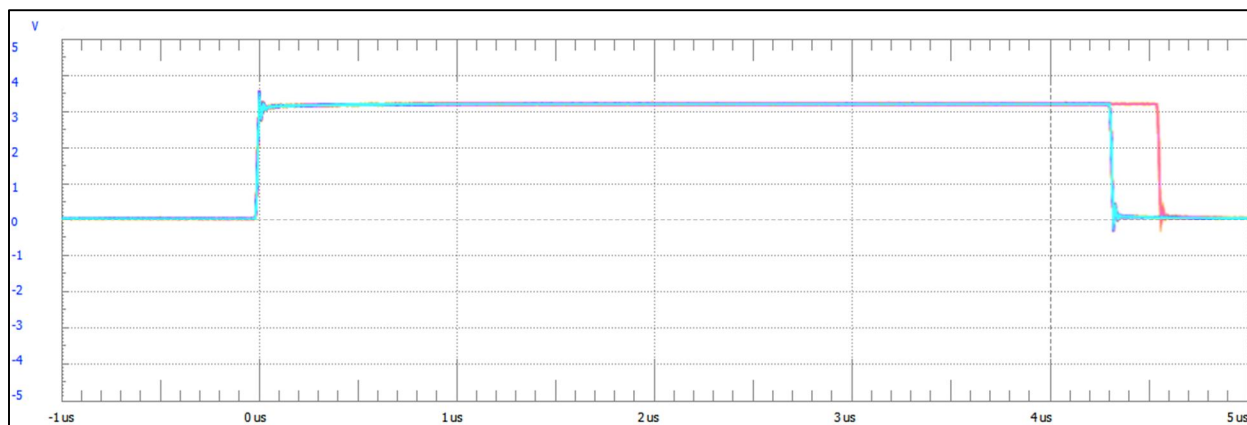


Рис. 4. Оциллограмма перекрёстно замкнутых TX и RX линий в момент отправки сообщения по протоколу UART

Fig. 4. Oscillogram of cross-connected TX and RX lines at the moment of sending a message via the UART protocol

Дребезг контактов – искажение сигнала, которое возникает при использовании механических контактов, когда кон-

такты многократно открываются и закрываются на коротких промежутках времени. Он может привести к непред-

сказуемым изменениям сигналов и прерываниям в передаче данных, что может повлиять на качество связи.

Для борьбы с дребезгом контактов можно использовать как аппаратные фильтры, так и программные алгоритмы фиксации и использование стабильных и надежных механических контак-

тов. На практике, подобные скорости в связке с БПЛА недостижимы на текущий момент времени и уровне развития информационных технологий. Проведя практические испытания в ходе исследования, была составлена таблица практической оценки протоколов потоковой передачи (табл.5):

Таблица 5. Сравнение протоколов поточной передачи видео по надёжности

Table 5. Comparison of video streaming protocols by reliability

№	Протокол / Protocol	Оценка надёжности / Reliability Assessment	Обоснование / Justification /
1	SRT	Высокая	SRT обладает механизмами для обнаружения и восстановления потерянных пакетов данных, что делает его надежным в условиях сетевых неполадок и потерь. Поддерживает адаптивную потоковую передачу, которая позволяет приспосабливаться к изменениям в сети и обеспечивать стабильную передачу данных
2	WebRTC	Высокая	WebRTC использует прямую пиринговую коммуникацию, что минимизирует задержки и уменьшает зависимость от серверов. Предоставляет механизмы для обнаружения и восстановления потерянных пакетов, что делает его надежным в условиях нестабильных сетей
3	RTP	Высокая	RTP предоставляет низкоуровневые механизмы транспортировки данных и обычно используется совместно с другими протоколами, такими как RTCP (Real-Time Control Protocol), что позволяет обнаруживать потерянные пакеты и восстанавливать поток данных
4	RTSP	Средняя	RTSP предназначен для управления потоками данных, а не для их транспортировки. Он обычно используется с другими протоколами, такими как RTP, для передачи мультимедиа данных. Надежность RTSP зависит от выбранного транспортного протокола (например, RTP) и его механизмов обнаружения ошибок

Для передачи служебной информации требования к сети гораздо ниже, при анализе учитывается, что:

1. Размер пакета Heartbeat и информации о датчиках погоды составляет 100 байт.

2. Средний размер команд MAVLink и MQTT прием равным 50 байт.

3. Частота отправки Heartbeat и данных о датчиках с БПЛА на землю - 1 раз в секунду.

4. Частота отправки команд с земли на БПЛА – 30 команд в секунду.

5. Объем данных от команд с земли на БПЛА прием равным 1500 байт в секунду (30 команд * 50 байт).

Проведя теоретические расчёты, получаем:

– С БПЛА на землю: объем данных Heartbeat и информация о датчиках: $(100 \text{ байт/сек} * 8 \text{ бит/байт}) / (1 \text{ Кбит/сек}) = 0,8 \text{ Кбит/сек}$.

– С земли на БПЛА: $(1500 \text{ байт/сек} * 8 \text{ бит/байт}) / (1 \text{ Кбит/сек}) = 12 \text{ Кбит/сек}$.

Таким образом, для передачи данных с БПЛА на землю требуется примерно 0,8 Кбит/сек интернет-трафика, а для передачи команд с земли на БПЛА требуется примерно 12 Кбит/сек интернет-трафика, не учитывая задержку в сети и возможную потерю пакетов.

В ходе анализа объема интернет-трафика, переданных данных управления БПЛА и информации о датчиках с учетом задержки, скорости передачи, потери данных и снижения скорости трафика при наборе скорости дроном удалось составить следующую таблицу (табл. 6).

С поправками: Задержка (latency) - 100 миллисекунд (0,1 секунды) в одну сторону (one-way latency); Скорость передачи (throughput) - 1,5 Мбит/сек (пропускная способность с земли на БПЛА).

Таблица 6. Объем интернет-трафика для данных управления и информации о датчиках

Table 6. Internet traffic volume for control data and sensor information

Скорость движения (км/ч) / Driving speed (km/h)	Задержка (в секундах) / Delay (in seconds)	Скорость соединения с учетом уменьшения (Мбит/сек) / Connection speed taking into account the reduction (Mbit/sec)	Размер данных с учетом задержки (в Кбит) / Data size with delay (in Kbit)	Время передачи (в секундах) / Transmission time (in seconds)
0	0.1	1,50	0.2	0.1
20	0.1	1.485	0.2	0.135
40	0.1	1.477	0.2	0.135
60	0.1	1.455	0.2	0.135
72	0.1	1.455	0.2	0.108
80	0.1	1,44	0.2	0.108
100	0.1	1.425	0.2	0.108

Ввиду небольшого объема данных задержка сети незначительно влияет на передачу команд БПЛА, что на больших скоростях не причинило неудобств в тестовых испытаниях.

Выводы

С помощью серии экспериментов и симуляций в статье оценивается эффективность предложенного подхода по снижению уязвимостей и смягчению потенциальных рисков, связанных с обменом данными в БПЛА. Результаты демонстрируют, что включение интеллектуальных компонентов в процесс обмена данными значительно повышает

общую безопасность информации, минимизируя нагрузку на системы БПЛА. Данное исследование вносит вклад в совершенствование безопасности информации для БПЛА, предоставляя практические знания по оптимизации протоколов обмена данными с использованием интеллектуальных компонентов. Полученные результаты окажут пользу операторам БПЛА, производителям и специалистам по безопасности, способствуя разработке более надежных и безопасных систем БПЛА, что является важным для безопасности и надежности их применения в современных приложениях.

Список литературы

1. Черный С. Г., Шипаренко Н., Чупаков М. Оценка нагруженности протоколов обмена данными для БПЛА на базе интеллектуальных компонентов // Известия Юго-Западного государственного университета. 2023. Т. 27, №3. С. 128-151. <https://doi.org/10.21869/2223-1560-2023-27-3-128-151>.
2. Аблец А. А., Стребков А. Н., Завгородняя Е. В. Опыт создания роя БПЛА в вооруженных силах иностранных государств // Военная мысль. 2022. № 6. С. 137-146. EDN ISYTIR.
3. Пенской И. С., Рогозин О. В. Нейро-нечеткий ПИД-регулятор в задаче угловой стабилизации мультироторного БПЛА // Новые информационные технологии в автоматизированных системах. 2018. № 21. С. 320-327. EDN XNRSJV.
4. Актуальные вопросы управления группой БПЛА / И. Н. Балашенко, Н. А. Горыня, А. С. Гунько, В. И. Никонов // Динамика систем, механизмов и машин. 2022. Т. 10, № 1. С. 14-17. <https://doi.org/10.25206/2310-9793-2022-10-1-14-17>. EDN XHENPM.
5. Подоплекин Ю. Ф., Толмачев С. Г., Шаров С. Н. Информационно-управляющая система приведения беспилотных летательных аппаратов на движущееся судно // Информационно-управляющие системы. 2012. № 3(58). С. 22-28. EDN OYIGWZ.
6. Силкин А. А., Ивановский А. Н., Черный С. Г. Автоматизированная система управления БПЛА в пределах локальной системы позиционирования // Состояние и перспективы развития современной науки по направлению "Робототехника": сборник

статей IV Всероссийской научно-технической конференции, Анапа, 20–21 июля 2022 года. Анапа: Федеральное государственное автономное учреждение "Военный инновационный технополис "ЭРА", 2022. С. 224-235. EDN WTLPIU.

7. Michels M., Wever H., Musshoff O. 'Drones take flight in forests: Uncovering the 'tree'-mendous potential and 'timber'-rific challenges for German forest managers by applying the technology acceptance model' // *Forest Policy and Economics*, 2023, № 157. P. 103077. <https://doi.org/10.1016/j.forpol.2023.103077>.

8. Zhu Z., Jeelani I., Gheisari M. 'Physical risk assessment of drone integration in construction using 4D simulation' // *Automation in Construction*. 2023. № 156. P. 105099. <https://doi.org/10.1016/j.autcon.2023.105099>.

9. Garg V. et al. 'Drones in last-mile delivery: A systematic review on efficiency, accessibility, and Sustainability' // *Transportation Research Part D: Transport and Environment*. 2023. № 123. P. 103831. <https://doi.org/10.1016/j.trd.2023.103831>.

10. Кутахов В. П., Мещеряков Р. В. Управление групповым поведением беспилотных летательных аппаратов: постановка задачи применения технологий искусственного интеллекта // *Проблемы управления*. 2022. № 1. С. 67-74. <https://doi.org/10.25728/ru.2022.1.5>. EDN J LXUXI.

11. Мещеряков Р. В., Илюхин Б. В. Обзор соревновательной робототехники. Соревнования спортивных роботов // *Робототехника и техническая кибернетика*. 2022. Т. 10, № 4. С. 255-260. <https://doi.org/10.31776/RTCJ.10402>. EDN ISXMEV.

12. Емельянов С.Г., Бобырь М. В., Бондаренко Б. А. Нечетко-логическая система распознавания цвета с помощью быстродействующего дефаззификатора // *Известия Юго-Западного государственного университета*. 2022. Т.26, № 4. С. 103-116. <https://doi.org/10.21869/2223-1560-2022-26-4-103-116>.

13. Емельянов С. Г., Бобырь М. В., Крюков А. Г. Исследование свойств алгоритма поиска в ширину для нахождения маршрута передвижения роботов // *Известия Юго-Западного государственного университета*. 2022. Т.26, №4. С. 39-56. <https://doi.org/10.21869/2223-1560-2022-26-4-39-56>.

14. Alwateer M., Loke S. W., Fernando N. Enabling drone services: Drone crowdsourcing and drone scripting // *IEEE Access*. 2019. Vol. 7. P. 110035-110049. <https://doi.org/10.1109/ACCESS.2019.2933234>. EDN ICGTGY.

15. Hassija V., Saxena V., Chamola V. Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory // *Computer Communications*. 2020. Vol. 149. P. 51-61. <https://doi.org/10.1016/j.comcom.2019.09.021>. EDN WAIQEY.

16. Hu J., Lanzon A. An innovative tri-rotor drone and associated distributed aerial drone swarm control // *Robotics and Autonomous Systems*. 2018. Vol. 103. P. 162-174. <https://doi.org/10.1016/j.robot.2018.02.019>. EDN MEFVVB.

17. Kim H., Chang W. Intuitive Drone Control using Motion Matching between a Controller and a Drone // Archives of Design Research. 2022. Vol. 35, no. 1. P. 93-113. <https://doi.org/10.15187/adr.2022.02.35.1.93>. EDN RTZJES.

18. Cocomo-Ortega J. A., Martinez-Carranza J. A compact CNN approach for drone localisation in autonomous drone racing // Journal of Real-Time Image Processing. 2022. Vol. 19, no. 1. P. 73-86. <https://doi.org/10.1007/s11554-021-01162-3>. EDN CFYATS.

19. An Optimization Approach to Minimize the Expected Loss of Demand Considering Drone Failures in Drone Delivery Scheduling / M. Torabbeigi, G. J. Lim, N. Ahmadian, S. J. Kim // Journal of Intelligent and Robotic Systems. 2021. Vol. 102, no. 1. <https://doi.org/10.1007/s10846-021-01370-w>. EDN RBLKPP.

20. Noise Attenuation on IMU Measurement for Drone Balance by Sensor Fusion / M. L. Hoang, M. Carratu, V. Paciello, A. Pietrosanto // Conference Record - IEEE Instrumentation and Measurement Technology Conference, Virtual, Glasgow, 17–20 May 2021 года. Virtual, Glasgow, 2021. P. 9460041. <https://doi.org/10.1109/I2MTC50364.2021.9460041>. EDN HYQCOI.

References

1. Cherny S. G., Shiparenko N., Chupakov M. Assessment of Data Exchange Protocol Workload in Uavs on Intelligent Components. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2023; 27(3): 128-151 (In Russ.). <https://doi.org/10.21869/2223-1560-2023-27-3-128-151>

2. Ablec A. A., Strebkov A. N., Zavgorodnjaja E. V. Opyt sozdaniya roja BPLA v vooruzhennyh silah inostrannyh gosudarstv [The experience of creating a swarm of UAVs in the armed forces of foreign states]. *Voennaja mysl'*, 2022; (6): 137-146. (In Russ.). EDN ISYTIR.

3. Penskoj I. S., Rogozin O. V. Neuro-fuzzy PID controller in the problem of angular stabilization of a multicopter UAV. *Novye informacionnye tehnologii v avtomatizirovannyh sistemah = New information technologies in automated systems*, 2018; (2): 320-327. (In Russ.). EDN XNRSJV.

4. Balashenko I. N., Gorynja N. A., Gun'ko A. S., Nikonov V. I. Topical issues of UAV group management. *Dinamika sistem, mehanizmov i mashin = Dynamics of systems, Mechanisms and Machined*. 2022; 10 (1): 14-17. (In Russ.). <https://doi.org/10.25206/2310-9793-2022-10-1-14-17>. EDN XHEHPM.

5. Podoplekin Ju. F., Tolmachev S. G., Sharov S. N. Information and control system for bringing unmanned aerial vehicles to a moving vessel. *Informacionno-upravljajushhie sistemy = Information and Control Systems*. 2012; (3):22-28. (In Russ.). EDN OYIGWZ.

6. Silkin A. A., Ivanovskij A. N., Chernyj S. G. Automated UAV control system within the local positioning system. In: *Sostojanie i perspektivy razvitija sovremennoj nauki po napravleniju "Robototekhnika". Sbornik statej IV Vserossijskoj nauchno-tehnicheskoy konfer-*

encii = The state and prospects of development of modern science in the field of Robotics: Collection of articles of the IV All-Russian Scientific and Technical Conference. Anapa: Voennyj innovacionnyj tehnopolis "JeRA"; 2022. P. 224-235. (In Russ.). EDN WTLPIU.

7. Michels M., Wever H., Musshoff O. Drones take flight in forests: Uncovering the ‘tree’-mendous potential and ‘timber’-rific challenges for German forest managers by applying the technology acceptance model. *Forest Policy and Economics*. 2023; (157): 103077. <https://doi.org/10.1016/j.forpol.2023.103077>.

8. Zhu Z., Jeelani I., Gheisari M. Physical risk assessment of drone integration in construction using 4D simulation. *Automation in Construction*. 2023; (156): 105099. <https://doi.org/10.1016/j.autcon.2023.105099>.

9. Garg V. et al. Drones in last-mile delivery: A systematic review on efficiency, accessibility, and Sustainability. *Transportation Research Part D: Transport and Environment*. 2023; (123): 103831. <https://doi.org/10.1016/j.trd.2023.103831>.

10. Kutahov V. P., Meshherjakov R. V. Management of group behavior of unmanned aerial vehicles: setting the task of applying artificial intelligence technologies. *Problemy upravlenija = Problems of Management*. 2022; (1): 67-74. (In Russ.). <https://doi.org/10.25728/pu.2022.1.5>. EDN JLXUXI.

11. Meshherjakov R. V., Iljuhin B. V. Review of competitive robotics. Combinations of sports robots. *Robototekhnika i tehničeskaja kibernetika = Robotics and Technical Cybernetics*. 2022; 10 (4): 255-260. <https://doi.org/10.31776/RTCJ.10402>. (In Russ.). EDN ISXMEV.

12. Emelianov S. G., Bobyr M. V., Bondarenko B. A. Fuzzy-logic Color Recognition System Using a Fast Defuzzifier. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2022; 26(4): 103-116 (In Russ.). <https://doi.org/10.21869/2223-1560-2022-26-4-103-116>.

13. Emelianov S. G., Bobyr M. V., Kryukov A. G. Research of the Properties of the Breadth-First Search Algorithm for Finding the Movement Route of Robots. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2022; 26(4): 39-56 (In Russ.). <https://doi.org/10.21869/2223-1560-2022-26-4-39-56>.

14. Alwateer M., Loke S. W., Fernando N. Enabling drone services: Drone crowdsourcing and drone scripting. *IEEE Access*. 2019; (7): 110035-110049. <https://doi.org/10.1109/ACCESS.2019.2933234>. EDN ICGTGY.

15. Hassija V., Saxena V., Chamola V. Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory. *Computer Communications*. 2020; 149:51-61. <https://doi.org/10.1016/j.comcom.2019.09.021>. EDN WAIQEY.

16. Hu J., Lanzon A. An innovative tri-rotor drone and associated distributed aerial drone swarm control. *Robotics and Autonomous Systems*. 2018; 103: 162-174. <https://doi.org/10.1016/j.robot.2018.02.019>. EDN MEFVVB.

17. Kim H., Chang W. Intuitive Drone Control using Motion Matching between a Controller and a Dron. *Archives of Design Research*. 2022; 35 (1): 93-113. <https://doi.org/10.15187/adr.2022.02.35.1.93>. EDN RTZJES.
18. Cocoma-Ortega J. A., Martinez-Carranza J. A compact CNN approach for drone localisation in autonomous drone racing. *Journal of Real-Time Image Processing*. 2022; 19 (1): 73-86. <https://doi.org/10.1007/s11554-021-01162-3>. EDN CFYATS.
19. Torabbeigi M., Lim G. J., Ahmadian N., Kim S. J. An Optimization Approach to Minimize the Expected Loss of Demand Considering Drone Failures in Drone Delivery Scheduling. *Journal of Intelligent and Robotic Systems*. 2021; 102 (1). <https://doi.org/10.1007/s10846-021-01370-w>. EDN RBLKPP.
20. Hoang M. L., Carratu M., Paciello V., Pietrosanto A. Noise Attenuation on IMU Measurement for Drone Balance by Sensor Fusion. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*. Virtual, Glasgow, 2021. 9460041 p. <https://doi.org/10.1109/I2MTC50364.2021.9460041>. EDN HYQCOI.

Информация об авторах / Information about the Authors

Черный Сергей Григорьевич, кандидат технических наук, доцент кафедры электрооборудование судов и эксплуатации производства, профессор кафедры КОИБ, Государственный университет морского и речного флота имени адмирала С.О. Макарова, г. Санкт-Петербург, Российская Федерация, Керченский государственный морской технологический университет, г. Керчь, Российская Федерация, e-mail: sergiiblack@gmail.com

Sergey G. Cherny, Cand. of Sci. (Engineering), Associate Professor of Electrical Equipment of Ships and Production Operation Department, Professor of the Department of COIB, State University of Marine and River Fleet named after admiral S.O. Makarov, St. Petersburg, Russian Federation, Kerch State Marine Technological University, Kerch, Russian Federation, e-mail: sergiiblack@gmail.com

Зонтова Татьяна Владимировна, доктор технических наук, профессор кафедры боевого применения радиотехнических средств, Черноморское высшее военно-морское училище имени П.С. Нахимова, г. Севастополь, Российская Федерация, e-mail: zontova_tatyana@mail.ru

Tatyana V. Zontova, Dr. of Sci. (Engineering), Professor of the of Combat Use of Radio-Technical Mean Department, Electrical Equipment Black Sea Higher Naval School named after P.S. Nakhimov, Sevastopol, Russian Federation, e-mail: zontova_tatyana@mail.ru

Шапаренко Никита Витальевич, студент, Государственный университет морского и речного флота имени адмирала С.О. Макарова, г. Санкт-Петербург, Российская Федерация, e-mail: nikita4shap@gmail.com

Nikita V. Shaparenko, Student, State University of Marine and River Fleet named after admiral S.O. Makarov, St. Petersburg, Russian Federation, e-mail: nikita4shap@gmail.com