



РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОТОКОЛА ДЛЯ ИНФОРМАЦИОННОГО ОБМЕНА МЕЖДУ УСТРОЙСТВАМИ УНУ АССИСТ

Н. И. Синёв, Д. И. Яцкевич, Д. М. Астахов, И. М. Воронов

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В статье описан протокол передачи данных в сети, основанный на стандарте SpaceWire. Этот протокол может использоваться для разработки и тестирования сетей SpaceWire. Описан стенд, используемый для разработки и тестирования протокола, его узлы и модули, их соединение, стандарт SpaceWire, а также приведена информация по разработанному протоколу тестирования сети – форматы пакетов, полей пакета, алгоритм передачи пакетов между узлами сети.

Ключевые слова: SpaceWire, бортовые сети, протоколы передачи данных, разработка бортовых сетей, тестирование бортовых сетей, лабораторный стенд.

Для цитирования:

Синёв, Н. И. Разработка и реализация протокола для информационного обмена между устройствами УНУ АССИСТ / Н. И. Синёв, Д. И. Яцкевич, Д. М. Астахов, И. М. Воронов // Системный анализ и логистика. – 2023. – № 3(37). – с. 104 – 121. DOI: 10.31799/2077-5687-2023-3-104-121.

DEVELOPMENT AND IMPLEMENTATION OF A PROTOCOL FOR INFORMATION EXCHANGE BETWEEN USI ASSRCT DEVICES

N. I. Sinyov, D. I. Yatskevich, D. M. Astakhov, I. M. Voronov

St. Petersburg State University of Aerospace Instrumentation

The article describes a data transmission protocol in a network based on the SpaceWire standard. This protocol can be used to develop and test SpaceWire networks. The stand used for the development and testing of the protocol, its nodes and modules, their connection, the SpaceWire standard are described, as well as information on the developed network testing protocol – packets format, packet fields, packet transmission algorithm between network nodes.

Keywords: SpaceWire, on-board networks, data transmission protocols, on-board network development, on-board network testing, laboratory stand.

For citation:

Sinyov, N. I. Development and implementation of a protocol for information exchange between USI ASSRCT devices / N. I. Sinyov, D. I. Yatskevich, D. M. Astakhov, I. M. Voronov // System analysis and logistics. – 2023. – № 3(37). – p. 104 – 121. DOI: 10.31799/2077-5687-2023-3-104-121.

Введение

В настоящем люди все больше уделяют внимание космическим исследованиям и разработкам космических систем и аппаратов. Как и любая система, космический аппарат состоит из подсистем – модулей. Одной из задач проектирования этих систем состоит в организации этих модулей в единую бортовую сеть, выработка стандартов их взаимодействия. Есть множество стандартов спутниковых бортовых сетей: MIL-STD-1553[1], SpaceWire[1, 2], TTEthernet[3] и т. д. На данный момент в бортовые сети космических аппаратов активно внедряется стандарт SpaceWire, и поэтому встают задачи разработки и тестирования сетей, основанных на этом стандарте. Для их решения нужно разработать универсальный протокол, работающий поверх сети SpaceWire и реализующий информационный обмен данными и конфигурацию сети. Поскольку протокол универсален, его можно проектировать и тестировать на специальном лабораторном стенде, а не на прототипе космического аппарата.

Реализация тестового стенда описана в [4]. Данная установка будет использована как рабочая сеть, на которой будут основаны тесты разрабатываемого протокола.



1. Описание системы для тестирования сети SpaceWire и его модулей [4]

Для тестирования сети SpaceWire используется уникальная научная установка аэрокосмический стенд SpaceWire для исследования, сертификации, тестирования (УНУ АССИСТ) (стенд).

На рисунке 1 изображена общая блок-схема УНУ АССИСТ. На рисунке 2 – его внешний вид.

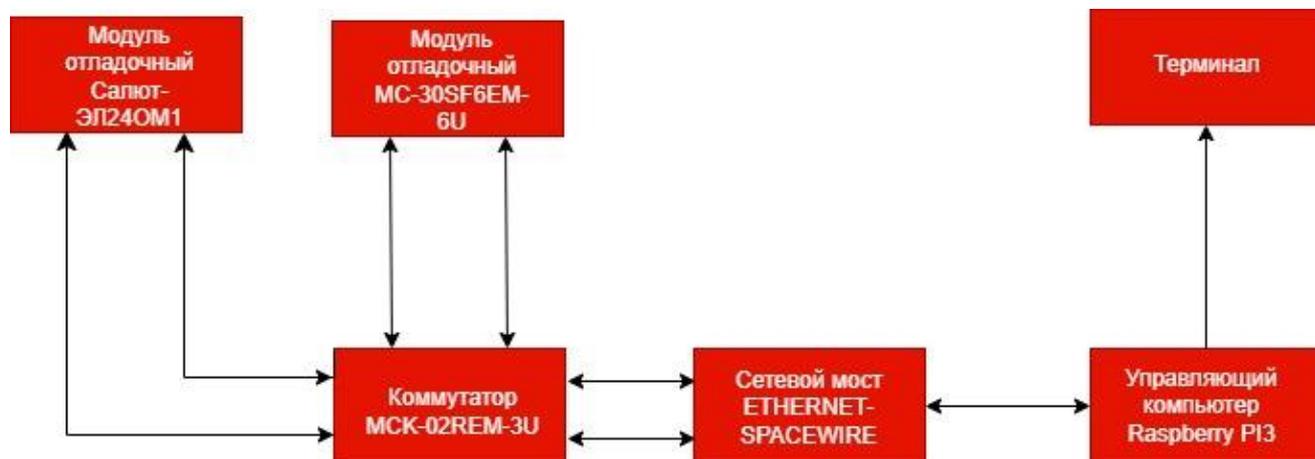


Рис. 1. Блок-схема УНУ АССИСТ



Рис. 2. Внешний вид УНУ АССИСТ



УНУ АССИСТ состоит из:

- управляющего компьютера Raspberry Pi3(RPi3), изображен на рисунке 3;
- терминала (ЖК монитора), изображен на рисунке 4;
- сетевого моста Ethernet-SpaceWire, изображен на рисунке 5;
- коммутатора SpaceWire МКК-02РЕМ-3U, изображен на рисунке 6;
- вычислительного модуля Салют-ЭЛ24ОМ1, изображен на рисунке 7;
- вычислительного модуля МС-30SF6ЕМ-6U, изображен на рисунке 8;



Рис. 3. Внешний вид управляющего компьютера Raspberry Pi3

```
processor           : 2
model name         : ARMv7 Processor rev 4 (v7l)
BogoMIPS          : 30.40
Features           : half thumb fastmult vfp edsp neon vfpv3 tlx vfpv4 idintv dinst of
p332 lpae evtstrm crc32
CPU implementer   : 0x41
CPU architecture  : 7
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4

processor           : 3
model name         : ARMv7 Processor rev 4 (v7l)
BogoMIPS          : 30.40
Features           : half thumb fastmult vfp edsp neon vfpv3 tlx vfpv4 idintv dinst of
p332 lpae evtstrm crc32
CPU implementer   : 0x41
CPU architecture  : 7
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4

Hardware          : BCM2085
Revision         : a02082
Serial           : 000000000002b4c25f
pi@raspberrypi:~$
```

Рис. 4. Внешний вид терминала



Рис. 5. Внешний вид сетевого моста Ethernet-SpaceWire



Рис. 6. Внешний вид коммутатора SpaceWire MCK-02REM-3U

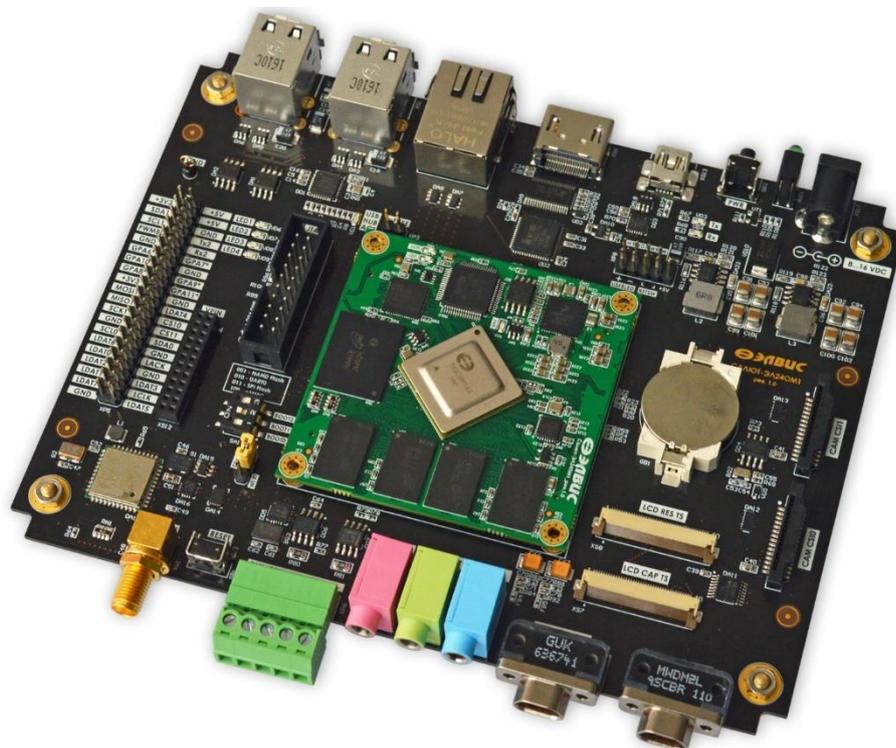


Рис. 7. Внешний вид вычислительного модуля Салют-ЭЛ24ОМ1

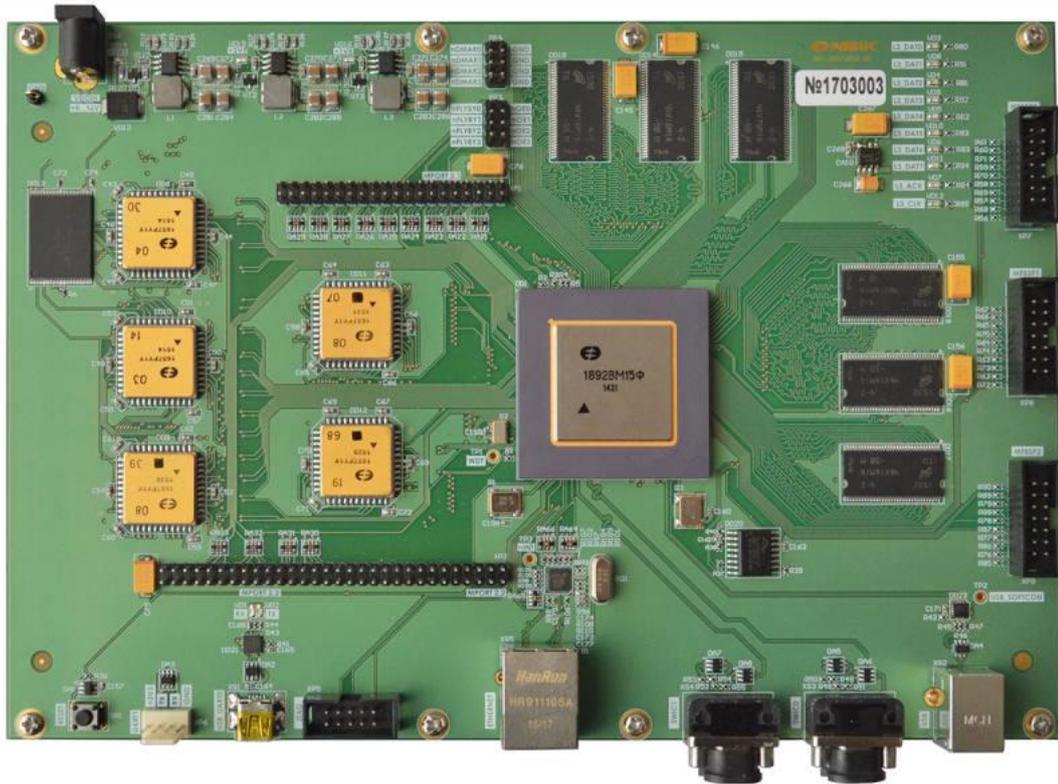


Рис. 8. Внешний вид вычислительного модуля MC-30SF6EM-66U

На управляющем компьютере Raspberry Pi3 установлена операционная система Linux, так как она является наиболее оптимальной ввиду относительно невысокой производительности компьютера. Также на нем содержатся утилиты, предназначенные для взаимодействия RPi3 с коммутатором, мостом и вычислительными модулями. С ним через HDMI-интерфейс соединен терминал, представленный ЖК-монитором. Сетевой мост обеспечивает соединение SpW-интерфейсов коммутатора с Ethernet-интерфейсом управляющего компьютера. Мост имеет один Ethernet-разъем, которым он соединен с RPi3 и два SpW-разъема, которыми он соединён с коммутатором.

Коммутатор представляет собой вычислительный модуль, определённым образом перенаправляющий поток данных с одного своего SpW-интерфейса на другой. Всего у него интерфейсов(разъёмов) SpaceWire – 8. Из них одна пара соединена с мостом, другая – с вычислительным модулем Салют-ЭЛ24ОМ1, третья – с вычислительным модулем MC-30SF6EM-66U. В свою очередь, эти два вычислительных модуля подобны RPi3 – они также представляют собой вычислительные блоки, содержащие ОС Linux и утилиты для взаимодействия с сетью SpaceWire, предназначенные для моделирования поведения бортовых модулей, имеющие, кроме всего прочего, пару интерфейсов SpaceWire, которыми они соединяются с коммутатором.

2. Описание стандарта SpaceWire[1]

SpaceWire – стандарт бортовых сетей, предназначенный для коммуникации между модулями летательного либо космического аппарата. Основной вид соединения – точка-точка, сеть строится на коммутаторах и может быть произвольной структуры. Поддерживаются скорости от 5 до 400 Мбит/с, в стандарте GigaSpaceWire – 625 Мбит/с, 1,25 Гбит/с и 2,5 Гбит/с [2]. Стек протоколов стандарта SpaceWire изображён на рисунке 9.

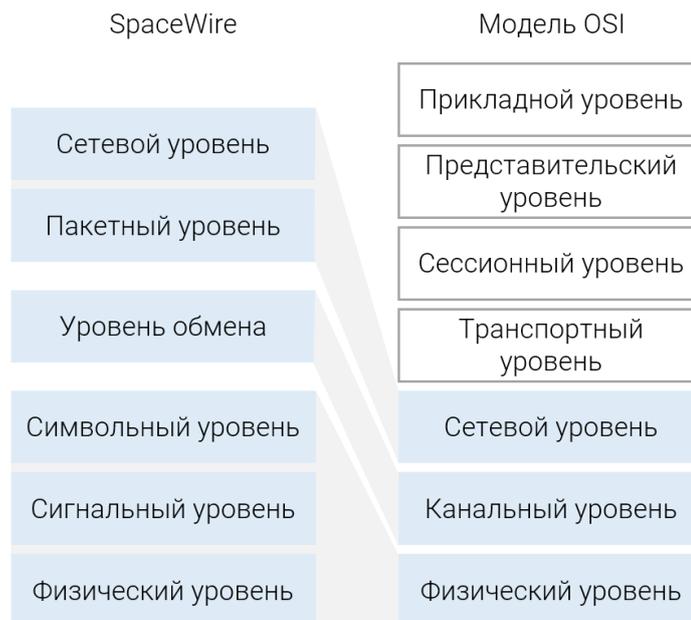


Рис. 9. Стек протоколов стандарта SpaceWire

Как видно на рисунке 9, только четыре самых нижних уровня в модели OSI имеют соответствие уровням протокола SpaceWire: физический уровень OSI соответствует физическому, сигнальному и символьному уровню в SpaceWire, уровень обмена в SpaceWire соответствует канальному уровню в OSI, сетевому уровню OSI соответствует уровень обмена и пакетный уровень в SpaceWire.

2.1. Физический уровень протокола SpaceWire

На этом уровне описываются физические требования к линии передачи. Структура кабеля SpaceWire представлена на рисунке 10.

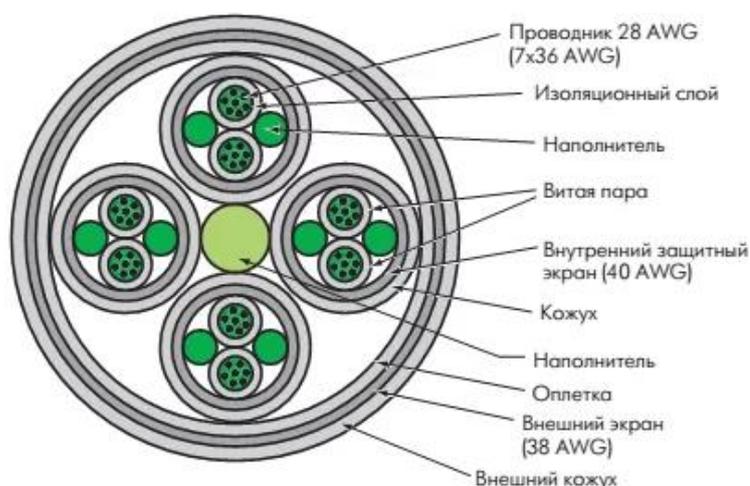


Рис. 10. Структура кабеля SpaceWire

Кабель состоит из четырёх витых пар с шагом скрутки, находящимся в пределах от 12 до 16 внешних диаметров неэкранированной витой пары. Одна витая пара есть два скрученных провода и два наполнителя витой пары, обёрнутые в защитный экран и кожух. Витые пары вместе с наполнителем кабеля обернуты оплёткой, экраном и внешним кожухом. Вес кабеля должен быть не более 80 г/м, длина не более 10 м. Разъем кабеля SpaceWire изображен на рисунке 11.

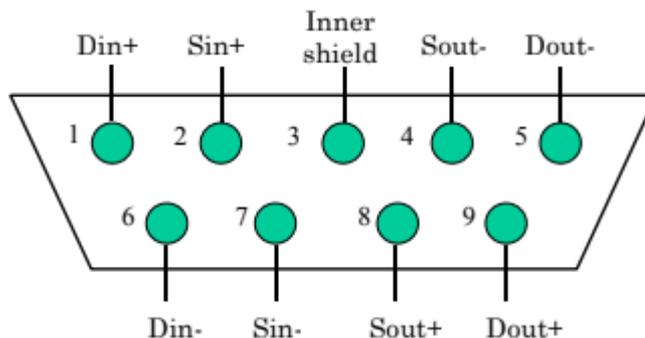


Рис. 11. Разъем кабеля SpaceWire [5]

Общий вид линии связи между двумя интерфейсами SpaceWire изображен на рис. 12.

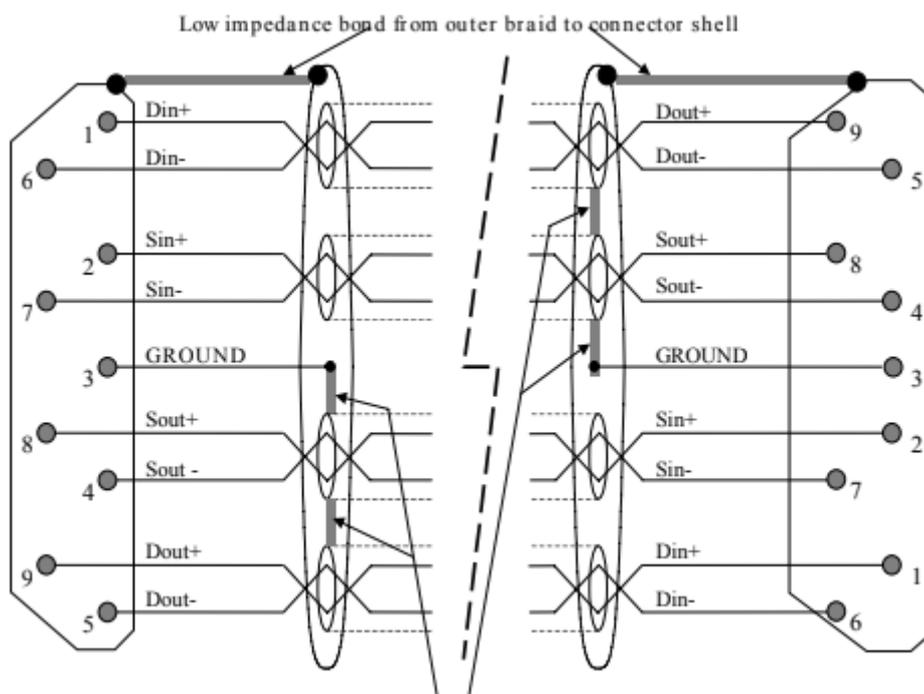


Рис. 12. Общий вид линии связи [5]

На рисунке 12 цифрой '1' обозначено соединение экрана кабеля и металлической оболочки разъема, цифрой '2' соединение выходных разъемов одного интерфейса с входными интерфейсами другого. Такое соединение снимает проблему определения стороны кабеля SpaceWire при сборке.

2.2. Сигнальный уровень

Каждая витая пара есть линия передачи данных. Их всего 4. Одна пара линий используется интерфейсом устройства для передачи данных, вторая – для приема. В паре линий, передающих в одном направлении, по одной линии передаётся сигнал данных (D-сигнал), по второй – строб-сигнал (S-сигнал), такой, что при его сложении по модулю два с сигналом данных получится тактирующий сигнал. Каждая витая пара проводит два дифференциальных сигнала величиной от 250 до 400 мВ относительно нуля. Эти сигналы на стороне передатчика генерируются выходным дифференциальным каскадом на транзисторах, а на приемной стороне подаются на вход дифференциального усилителя, входное дифференциальное сопротивление которого равно 100 Ом. Схема выходного и входного каскада и изображены на рисунке 13.

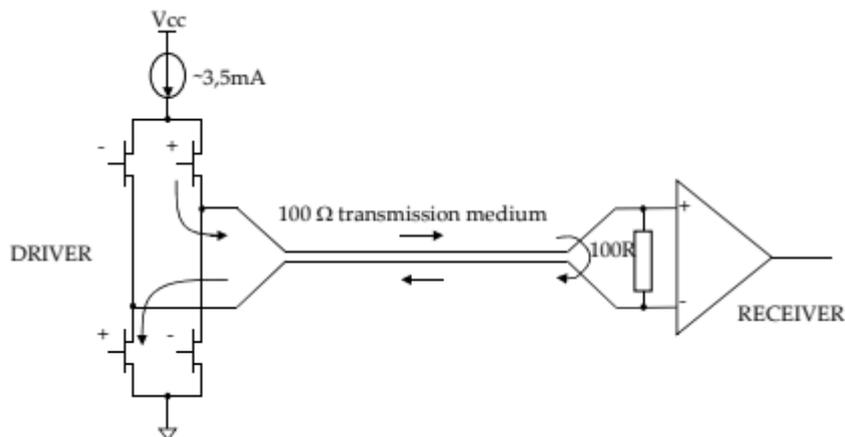


Рис. 13. Схема входного и выходного каскада интерфейсов SpaceWire и их соединения [5]

2.3. Символьный уровень

В стандарте определены следующие символы (для всех символов P – бит чётности):

1) Управляющие символы:

- 1.1) FCT-символ: P100; код контроля потока, используется для учета свободного места в буфере приемника;
- 1.2) EOP-символ: P101; код, находящийся в конце пакета и означающий корректную передачу пакета;
- 1.3) EEP-символ: P110; код, находящийся в конце пакета и означающий некорректную передачу пакета;
- 1.4) ESC-символ: P111, используется для обозначения управляющих кодов.

2) Управляющие коды:

- 2.1) Time-код: P11110t₀t₁t₂t₃t₄t₅t₆, где t₀-t₆ – значащие разряды Time-кода; эти коды используются для синхронизации времени модулей;
- 2.2) Interrupt-код: P11110i₀i₁i₂i₃i₄001, где i₀-i₄ – значащие разряды Interrupt-кода; эти коды, как видно из названия, представляют собой прерывания;
- 2.3) Interrupt acknowledge-код: P11110i₀i₁i₂i₃i₄101, где i₀-i₄ – значащие разряды Interrupt acknowledge-кода; эти коды есть ответы на прерывания;
- 3) Символ данных: P0d₀d₁d₂d₃d₄d₅d₆d₇, где d₀-d₇ – значащие разряды символа данных;

Для всех символов первым идёт P, затем – остальные биты, т. е., порядок поступления битов в канал – слева направо. Младший значащий бит обозначается нулевым индексом, старший, соответственно, максимальным. Расчет бита четности показан на рисунке 14.

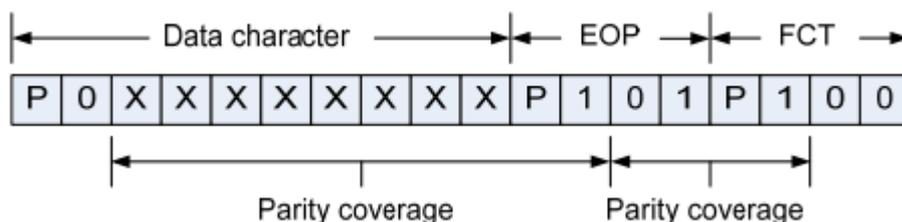


Рис. 14. Схема расчета бита четности [5]

Для символа данных бит четности считается по значащим битам текущего символа и двум битам следующего, для контрольных символов по двум последним битам текущего и двум битам следующего символов.



2.4. Уровень обмена

Стандарт определяет контроль потока данных с помощью FCT-символов для избежания переполнения буфера. При наличии в интерфейсе приёмника свободного места тот передаёт n FCT-символов передатчику, и последний увеличивает свой счетчик кредитов на $FCT_COUNT * n$, и, если это нужно, передает данные. Обычно FCT_COUNT равен 7; также счетчик кредитов не может быть больше MAX_FCT_COUNT , обычно равного 56. Запуск соединения происходит с помощью машины состояний, описанной в [2].

2.5. Уровень пакетов

Канальный пакет состоит из трех частей: последовательность адресов(байты), байты данных и EOP/EER. EER добавляется в пакет при какой-либо ошибке передачи.

2.6. Сетевой уровень

Сеть SpaceWire строится на коммутаторах. Пример сети изображён на рисунке 15.

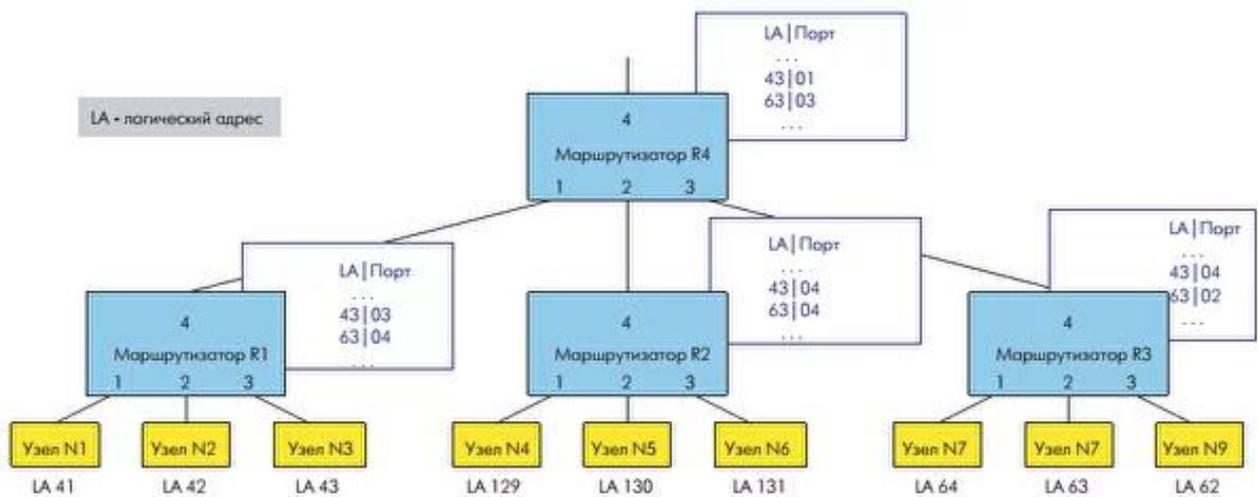


Рис. 15. Пример сети SpaceWire [6]

При вхождении пакета в коммутатор порт, в который должен отправиться пакет, определяется из первого байта, затем пакет передаётся. Адресный блок пакета формируется как последовательность портов/логических адресов, на которые маршрутизатор должен отправить пакет, причем пакет в конце концов должен прийти на целевое устройство.

В сетях используется два вида маршрутизации: путевая и логическая. При путевой адресации из первого байта читается номер выходного порта, и пакет передается без этого первого байта; при логической адресации пакет передается весь. Путевые адреса лежат в диапазоне от 1 до 31; логические – от 32 до 254, 0 – конфигурационный порт, 255 – зарезервирован. Маршрутизация осуществляется “на лету”: при поступлении пакета сразу определяется выходной порт и пакет сразу, без буферизации, передается. Такая маршрутизация еще называется “червячная”. Она изображена на рисунке 16.

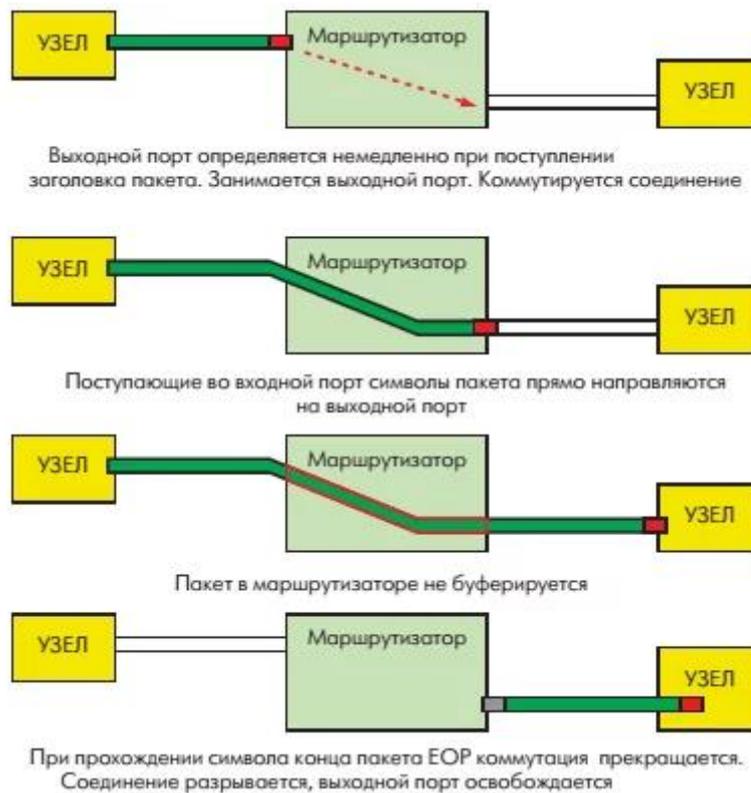


Рис 16. Червячная маршрутизация [6]

3. Описание протокола тестирования SpW-сети

3.1. Формат тестовых пакетов

Тестовые пакеты разделяются на шесть типов:

- Стартовый пакет;
- Подтверждение стартового пакета;
- Пакет начала теста;
- Пакет данных;
- Ответный пакет;
- Финальный пакет;

Форматы тестовых пакетов приведены на рисунках 17 – 20.

первый передаваемый байт

Адрес назначения	Идентификатор тестового пакета	Адресат обмена данными	Флаги
Номер теста	Адрес отправителя	Размер передаваемых данных	EOP

последний передаваемый байт

Рис. 17. Формат стартового пакета



первый передаваемый байт

Адрес назначения	Идентификатор тестового пакета	Флаги	Номер теста
Адрес отправителя	Статус	EOP	

последний передаваемый байт

Рис. 18. Формат ответного пакета, пакета начала теста и пакета подтверждения

первый передаваемый байт

Адрес назначения	Идентификатор тестового пакета	Флаги	Номер теста
Адрес отправителя	Длина поля данных	Данные	Данные
Данные	...	Данные	EOP

последний передаваемый байт

Рис. 19. Формат пакета данных

первый передаваемый байт

Адрес назначения	Идентификатор тестового пакета	Флаги	Номер теста
Адрес отправителя	Статус теста	EOP	

последний передаваемый байт

Рис. 20. Формат финального пакета

3.2. Описание служебных полей:

3.2.1. Поле 'Идентификатор тестового пакета'

1. Поле 'Идентификатор тестового пакета' содержит 8 бит, которые определяют идентификатор тестового пакета.
2. Для АССИСТ 'Идентификатор тестового пакета' принимается равным значению '240'.

3.2.2. Поле 'Флаги'

Поле 'Флаги' содержит 1 байт и определяет различные служебные флаги, необходимые для передачи пакетов в сети. Формат поля 'Флаги' приведен на рисунке 21.

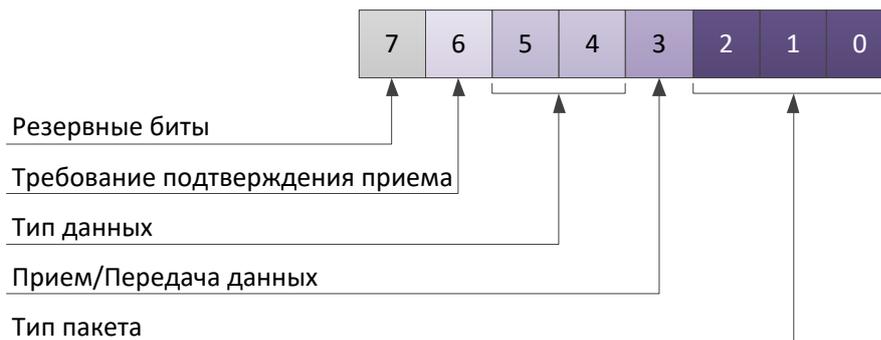


Рис. 21. Формат поля 'Флаги'

- 1) Поле «Тип пакета» содержит три бита и может принимать следующие значения:
 - '000'b – стартовый пакет;
 - '001'b – подтверждение стартового пакета;
 - '010'b – пакет начала теста.
 - '011'b – пакет данных;
 - '100'b – ответный пакет;
 - '101'b – финальный пакет;
- 2) Флаг «Прием/Передача данных» определяет направление передачи данных. Данный флаг используется только в стартовом пакете при конфигурации устройства. Может принимать следующие значения:
 - '0'b – прием данных;
 - '1'b – передача данных.
- 3) Поле «Тип данных» определяет тип передаваемых данных. Данное поле используется только в стартовом пакете при конфигурации устройства. Оно представлено двумя битами, которые могут принимать следующие значения:
 - '00'b – пакет данных;
 - '01'b – time-код;
 - '10'b – прерывание;
 - '11'b – зарезервировано для будущего использования.
- 4) Флаг «Требование подтверждения приема» определяет необходимость отправки пакета подтверждения приема данных. Данный флаг используется только в стартовом пакете при конфигурации устройства. Может принимать следующие значения:
 - '0'b – подтверждение не требуется;
 - '1'b – подтверждение требуется.
- 5) Младший бит поля 'Флаги' зарезервирован для будущего использования и должен быть установлен в ноль.

3.2.3. Поле 'Размер передаваемых данных'

- 1) Поле 'Размер передаваемых данных' используется в стартовых пакетах и предназначено для определения числа передаваемых пакетов и их размера.
- 2) Поле 'Размер передаваемых данных' содержит 1 байт и состоит из следующих полей:
 - Биты 0-3: «Длина поля данных». Содержит длину поля данных передаваемых пакетов. Длина поля данных передаваемых пакетов должна быть задана в соответствии с Таблица 1.



Таблица 1 – «Длина поля данных»

Длина поля данных	Значение
10 байт	'0001'b
100 байт	'0010'b
512 байт	'0011'b
1 Кбайт	'0100'b
2 Кбайт	'0101'b
4 Кбайт	'0110'b
16 Кбайт	'0111'b
32 Кбайт	'1000'b

- Биты 4-7: «Количество пакетов». Содержит количество передаваемых пакетов. Количество пакетов должно быть задано в соответствии с таблицей 2.

Таблица 2 – «Количество пакетов»

Количество пакетов	Значение
1 пакет	'0001'b
10 пакетов	'0010'b
100 пакетов	'0011'b
1000 пакетов	'0100'b
3000 пакетов	'0101'b

3.2.4. Поле 'Длина поля данных'

- 1) Поле 'Длина поля данных' используется в пакетах данных.
- 2) Поле 'Длина поля данных' содержит 1 байт и предназначено для определения длины поля данных пакета данных.
- 3) Биты 0-3 поля 'Длина поля данных' содержат длину поля данных. длина поля данных может быть задана в соответствии с Таблица 1.
- 4) Биты 4-7 поля 'Длина поля данных' зарезервированы для будущего использования.

3.2.5 Поле 'Адрес назначения'

Поле 'Адрес назначения' представлено одним байтом и может содержать:

- логический адрес узла назначения;
- путевой адрес узла назначения;

3.2.6 Поле 'Адресат обмена данными'

Поле 'Адресат обмена данными' содержит адрес узла, с которым узел назначения будет осуществлять обмен данными.

3.2.7 Поле 'Адрес отправителя'

Поле 'Адрес отправителя' представлено одним байтом и может содержать:

- логический адрес узла отправителя. В представлен перечень логических адресов устройств, входящих в состав АССИСТ;
- путевой адрес узла отправителя.

3.2.8 Поле 'Номер теста'

- 1) Поле 'Номер теста' представлено одним байтом;
- 2) Поле 'Номер теста' должно содержать номер теста.



3.2.9 Поле 'Статус'

- 1) Поле 'Статус' представлено одним байтом;
- 2) Поле 'Статус' устанавливается в значение '0' в случае успешного приема стартового пакета. В случае ошибки приема пакета стартового пакета, поле 'Статус' устанавливается в '1'.
- 3) Старшие 7 бит поля 'Статус' зарезервированы.

3.2.10 Поле 'Статус теста'

- 1) Поле 'Статус теста' используется только в финальных пакетах;
- 2) Поле 'Статус теста' представлено одним байтом;
- 3) Поле 'Статус теста' устанавливается в значение '0' в случае успешного завершения теста. В случае ошибки поле 'Статус теста' устанавливается в '1';
- 4) Старшие 7 бит поля 'Статус теста' зарезервированы.

Пакет должен быть завершен признаком конца пакета (EOP), который должен быть использован для указания нормального конца пакета.

3.3. Описание процесса обмена пакетами между вычислительными модулями уникальной научной разработки УНУ АССИСТ

- 1) Управляющий компьютер Raspberry Pi отправляет два стартовых пакета: один отладочному модулю Салют-ЭЛ24ОМ1, второй отладочному модулю МС-30SF6ЕМ-6U.
- 2) Отладочный модуль Салют-ЭЛ24ОМ1 принимает стартовые пакеты, устанавливает параметры конфигурации теста, указанные в этом пакете, формирует пакет ответа на стартовый пакет и отправляет его на управляющий компьютер Raspberry Pi3. Отладочный модуль МС-30SF6ЕМ-6U делает то же самое.
- 3) При получении двух пакетов ответа на стартовый пакет управляющий компьютер Raspberry Pi отправляет два пакета начала теста: один отладочному модулю Салют-ЭЛ24ОМ1, второй отладочному модулю МС-30SF6ЕМ-6U. Эти пакеты сигнализируют отладочным модулям о начале обмена данными между ними.
- 4) Отладочный модуль Салют-ЭЛ24ОМ1 и отладочный модуль МС-30SF6ЕМ-6U начинают обмен пакетами данных. Количество, размер пакетов, направление передачи данных (от Салют-ЭЛ24ОМ1 к МС-30SF6ЕМ-6U или наоборот), подтверждение приёма пакета данных определяются конфигурацией, данной в стартовом пакете.
- 5) При окончании обмена пакетами оба отладочных модуля шлют на управляющий компьютер Raspberry Pi3 финальные пакеты, содержащие статус теста.

3.4. Перечень тестовых сценариев

3.4.1. Режимы тестирования

Предусмотрены следующие режимы тестирования:

- Одиночный тест. Представляет собой запуск одного теста по выбору пользователя.
- Группа тестов. В данном режиме предусмотрены следующие группы тестов:
 - Тестирование совместимости с модулем Салют-ЭЛ24ОМ1. Представляет собой последовательный запуск тестов №1-60. Подробное описание тестов представлено в п. 2.4.2.
 - Тестирование совместимости с модулем МС-30SF6ЕМ-6U. Представляет собой последовательный запуск тестов №61-120.
 - Тестирование передачи данных. Представляет собой последовательный запуск тестов №1-15, №31-45, №61-75 и №91-115.



- Тестирование приема данных. Представляет собой последовательный запуск тестов №16-30, №46-60, №76-90 и №106-120.
- Быстрое тестирование. Представляет собой последовательный запуск тестов №3, 12, 18, 27, 33, 42, 48, 57, 63, 72, 78, 87, 93, 102, 108, 117.

3.4.2. Тестирование передачи пакетов данных из модуля МС-30SF6ЕМ-6U в модуль Салют-ЭЛ24ОМ1 (С-ЭЛ) на минимальной скорости

Пакеты данных передаются из модуля МС-30SF6ЕМ-6U в отладочный модуль Салют-ЭЛ24ОМ1 (С-ЭЛ). В таблице 3 представлены общие параметры передачи пакетов данных для всех тестовых сценариев, предназначенных для тестирования передачи пакетов данных на минимальной скорости.

Таблица 3 – Общие параметры тестов на передачу пакетов данных на минимальной скорости

Параметр	Значение
Маршрут передачи	МС-30 – С-ЭЛ
Скорость передачи	10 Мбит/с
Требование подтверждения приема	Подтверждение требуется(для тестовых сценариев №61-75 требуется)
Адресация	Путевая

В таблице 4 представлены параметры 15 тестовых сценариев, предназначенных для тестирования передачи пакетов данных на минимальной скорости. В соответствии с данными параметрами, а также параметрами, заданными в Таблица 4, МС-30SF6ЕМ-6U формирует пакеты данных.

Таблица 4 – Параметры тестовых сценариев

№ тестового сценария	Количество пакетов	Длина поля данных	Режим быстрого тестирования
	1	100 байт	
	1	512 байт	
	1	1 Кбайт	да
	1	2 Кбайт	
	1	4 Кбайт	
	1	16 Кбайт	
	1	32 Кбайт	
	10	10 байт	
	10	100 байт	
	10	512 байт	
	10	1 Кбайт	
	100	10 байт	да
	100	100 байт	
	1000	10 байт	
	3000	10 байт	

Такие параметры, как количество и размер пакетов для тестовых сценариев №61-75 идентичны соответствующим характеристикам тестовых сценариев №1-15.



3.4.3. Тестирование передачи пакетов данных из модуля Салют-ЭЛ24ОМ1 (С-ЭЛ) в отладочный модуль МС-30SF6ЕМ-6U (МС) на минимальной скорости

Пакеты данных передаются из модуля МС-30SF6ЕМ-6U в отладочный модуль Салют-ЭЛ24ОМ1 (С-ЭЛ). В таблице 5 представлены общие параметры передачи пакетов данных для всех тестовых сценариев, предназначенных для тестирования передачи пакетов данных на максимальной скорости.

Таблица 5 – Общие параметры тестов на передачу пакетов данных на максимальной скорости

Параметр	Значение
Маршрут передачи	МС-30SF6ЕМ-6U – С-ЭЛ
Скорость передачи	10 Мбит/с
Требование подтверждения приема	Подтверждение требуется
Адресация	Путевая

Такие параметры, как количество и размер пакетов для тестовых сценариев №16-30 и №76-90 идентичны соответствующим характеристикам тестовых сценариев №1-15.

3.4.4. Тестирование передачи пакетов данных из модуля МС-30SF6ЕМ-6U в модуль Салют-ЭЛ24ОМ1 (С-ЭЛ) на максимальной скорости

Пакеты данных передаются из модуля Салют-ЭЛ24ОМ1 (С-ЭЛ) в отладочный модуль МС-30SF6ЕМ-6U (МС).

В таблице 6 представлены общие параметры передачи пакетов данных для всех тестовых сценариев, предназначенных для тестирования передачи пакетов данных на минимальной скорости.

Таблица 6 – Общие параметры тестов на передачу пакетов данных на минимальной скорости

Параметр	Значение
Маршрут передачи	МС-30 – С-ЭЛ
Скорость передачи	200 Мбит/с
Требование подтверждения приема	Подтверждение не требуется(для тестов № 106-120 – требуется)
Адресация	Логическая

Такие параметры, как количество и размер пакетов для тестовых сценариев №31-45 и №106-120 идентичны соответствующим характеристикам тестовых сценариев №1-15.

3.4.5 Тестирование приема пакетов данных из модуля Салют-ЭЛ24ОМ1 (С-ЭЛ) в отладочный модуль МС-30SF6ЕМ-6U (МС) на максимальной скорости

Пакеты данных передаются из модуля Салют-ЭЛ24ОМ1 в отладочный модуль МС-30SF6ЕМ-6U (МС).

В таблице 7 представлены общие параметры передачи пакетов данных для всех тестовых сценариев, предназначенных для тестирования передачи пакетов данных на максимальной скорости.



Таблица 7 – Общие параметры тестов на передачу пакетов данных на максимальной скорости

Параметр	Значение
Маршрут передачи	Салют-ЭЛ240М1 – МС
Скорость передачи	200 Мбит/с
Требование подтверждения приема	Подтверждение не требуется(для тестовых сценариев №46-60 требуется)
Адресация	Логическая

Такие параметры, как количество и размер пакетов для тестовых сценариев №46-60 и №91-105 идентичны соответствующим характеристикам тестовых сценариев №1-15.

Заключение

В данной работе описан и реализован прототип протокола, позволяющий конфигурировать сеть УНУ АССИСТ и передавать данные между модулями данной установки. В статье описано устройство УНУ АССИСТ, на которой тестируется протокол, приведено краткое описание стандарта SpaceWire, и, затем, описана краткая спецификация разрабатываемого протокола. Разрабатываемый протокол включает в себя пакеты (один пакет данных и пять служебных), правила обмена этими пакетами и роли устройств в сети: одно(RPI3) начинает сессию передачи данных, второе(Салют-ЭЛ240М1) и третье(МС-30SF6ЕМ-GU) обмениваются данными. Сессия, или единичный тест обмена, описывает направление, скорость, наличие подтверждения передачи данных. Также протокол определяет тестовые сценарии – группы тестов, созданные по одному параметру (например, скорость или требование подтверждения).

Стандарт SpaceWire также включает в себя interrupt-коды, interrupt-acknowledge-коды и time-коды. В данном протоколе они не используются, но тесты, включающие их, могут быть добавлены. Также протокол может быть использован не только для передачи данных между модулями УНУ АССИСТ, но и для передачи данных между модулями реального космического аппарата. Программная реализация протокола проста в разработке, он включает возможность простой проверки наличия соединения между модулями. Но он не описывает детальную проверку пакетов на корректность, поведение модулей при получении некорректных пакетов или при их неполучении. Этот нереализованный функционал будет включен в будущую редакцию протокола.

СПИСОК ЛИТЕРАТУРЫ

1. Дымов. Д. В. Перспективы применения сетевой технологии SpaceWire на российских спутниках / Дымов Д. В., Двирный В. В., Ерёменко Н. В., Двирный Г. В. // Решетневские чтения. – 2015. – С. 103 – 105.
2. ГОСТ Р 70020-2022. Космическая техника. Интерфейсы и протоколы высокоскоростного межприборного информационного обмена и комплексирования бортовых систем космических аппаратов. SpaceWire-RUS. – Москва: Российский институт стандартизации, 2002. – 112 с.
3. Control Engineering Россия: Стандарт бортовой сети космической станции TTEthernet [Электронный ресурс]. – URL: <https://controlengrussia.com/promy-shlenny-e-seti/protokoly/ttethernet/> (дата обращения: 11.09.2023).
4. Синёв Н. И. Аппаратно-программное исследование коммуникационных протоколов для бортовых сетей / Синёв Н. И., Карандашев А. А., Оленев В. Л., Чумакова Н. Ю., Сыщиков А. Ю. // Системный анализ и логистика. – 2022. – № 1(31). – С 44-62.
5. SpaceWire Standard. Space Technology Centre. University of Dundee ECSS – Space Engineering. "SpaceWire – Links, Nodes, Routers and Networks". ECSS-E-ST-50-12 Draft H1: July 2008.



6. SpaceWire: Обзор технологии SpaceWire [Электронный ресурс]. – URL: <https://www.spacewire.ru/spacewire> (дата обращения: 13.10.2023).

ИНФОРМАЦИЯ ОБ АВТОРАХ

Синёв Николай Иванович –

Ст. преподаватель, мл. науч. сотрудник
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: nikolay.sinyov@guap.ru

Яцкевич Даниил Игоревич –

Студент
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: j.daniil@bk.ru

Астахов Даниил Максимович –

Студент
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: daniil-actaxov@yandex.ru

Воронов Илья Михайлович –

Студент кафедры аэрокосмических компьютерных программных систем
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: ilya.stud.spb@gmail.com

INFORMATION ABOUT THE AUTHORS

Sinev Nikolay Ivanovich –

Senior Lecturer, Associate Junior Research
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia
E-mail: nikolay.sinyov@guap.ru

Yatskevich Daniil Igorevich –

Student
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia
E-mail: j.daniil@bk.ru

Astakhov Daniil Maksimovich –

Student
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia
E-mail: daniil-actaxov@yandex.ru

Voronov Ilya Mikhailovich –

Student of the Departments of Aerospace Computer Software Systems
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia
E-mail: ilya.stud.spb@gmail.com