

ООО "Образование будущего"

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ «ОРБИТА.ЧЕЛЛЕНДЖ»**

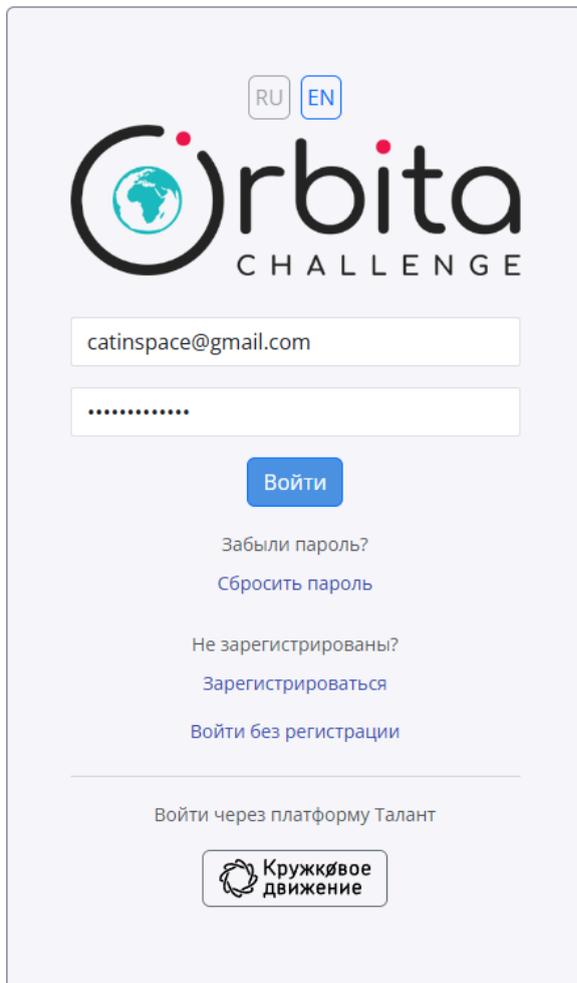
г. Москва
2024 г.

ОГЛАВЛЕНИЕ

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «ОРБИТА.ЧЕЛЛЕНДЖ».....	1
1. Начало работы.....	3
2. Отправка решений.....	3
3. Программа управления аппаратом.....	5
3.1. Классы и основы объектно-ориентированного программирования.....	5
3.2. Написание программы управления на языке JavaScript.....	8
3.3. Обзор устройств и функций космического аппарата.....	11
3.3.1. Аккумулятор.....	12
3.3.2. Солнечная панель.....	13
3.3.3. Маховик.....	14
3.3.4. Датчик угловой скорости.....	15
3.3.5. Навигационный датчик.....	16
3.3.6. Звездный датчик.....	17
3.3.7. Реактивный двигатель.....	17
3.3.8. Топливный бак.....	18
3.3.9. Передатчик.....	19
3.3.10. Приемник.....	20
3.3.11. Источник данных.....	21
3.3.12. Потребитель данных.....	22
3.3.13. Нагреватель.....	23
4. Активация лицензии (только для администраторов).....	23

1. Начало работы

Вход для всех пользователей осуществляется по ссылке: <https://orbita.education/>



После входа в систему пользователь попадает на страницу с доступными ему событиями.

2. Отправка решений

Решение задач

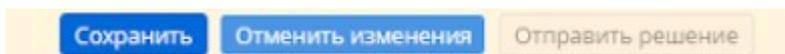
1. Выберите доступное вам событие.
2. На открывшейся странице будут краткие описания задач. В описаниях указаны сроки открытия и закрытия задач.
3. Кликните на название задачи, открытой на данный момент. Вам откроется описание задачи. Сверху над описанием есть кнопки в виде пятиугольных иконок. При клике на «Описание» отображается описание задачи; на «Начальные данные» - начальные условия, которые пригодятся при решении задачи; на «Критерии оценки» - критерии начисления баллов данной задачи; на «Решения» - открывается доступ к отправке решений.



Внимательно ознакомьтесь с **критериями оценки**.

Отправка решений

1. На вкладке «Решения» каждой задачи есть опция «Создать новое решение». При клике на надпись откроется страница с текстовыми полями, в которые необходимо ввести число или код программы.
2. При введении всех данных в верхнем правом углу нажмите «Сохранить»



3. Если вы уверены в правильности введенных данных нажмите в верхнем правом углу кнопку «Отправить»



4. Итог решения будет выведен после окончания моделирования.
5. Слева вы сможете увидеть результат всех ваших решений, а также скопировать их (при наведении на решение появляется иконка копирования).
6. У каждого решения также есть две пятиугольные иконки, показывающие введенные данные и результат.



3. Программа управления аппаратом

Перед решением задач участникам предлагается ознакомиться с API аппарата.

Аппарат в сервисе семейства "Орбита" состоит из т.н. устройств, представляющих единое целое с точки зрения проекта. Это может быть как отдельный датчик, так и целая подсистема. Устройство может поддерживать одну или несколько predetermined функций.

При создании аппарата вы можете добавить к любым устройствам любые функции. Например, вы можете оснастить спутник средствами связи. Для этого можно:

1. Добавить устройство "Радиомодуль" с тремя функциями - "Приемник", "Передатчик" и "Антенна", ил
2. Сделать два разных устройства с двумя функциями каждое, одно - только для приема, другое только для передачи, и у каждого будет своя антенна;
3. Сделать устройство с несколькими функциями "Передатчик", если оно может независимо работать на нескольких разных частотах
4. Добавить в устройство "Научный модуль" функцию "Передатчик", если по задумке разработчика это оборудование имеет свой радиомодуль и антенну.
5. И т.п.

3.1. Классы и основы объектно-ориентированного программирования

Для более детального понимания, как устроено API, следует ознакомиться с принципами объектно-ориентированного программирования.

В рамках этого подхода среди прочего объявляются так называемые типовые классы объектов, обладающими собственными свойствами (подчиненными переменными и объектами) и функциями. Некоторые из этих свойств и функций в программе могут быть доступны только для чтения, другие позволяют управлять этим объектом через вызовы его функций с определенными параметрами.

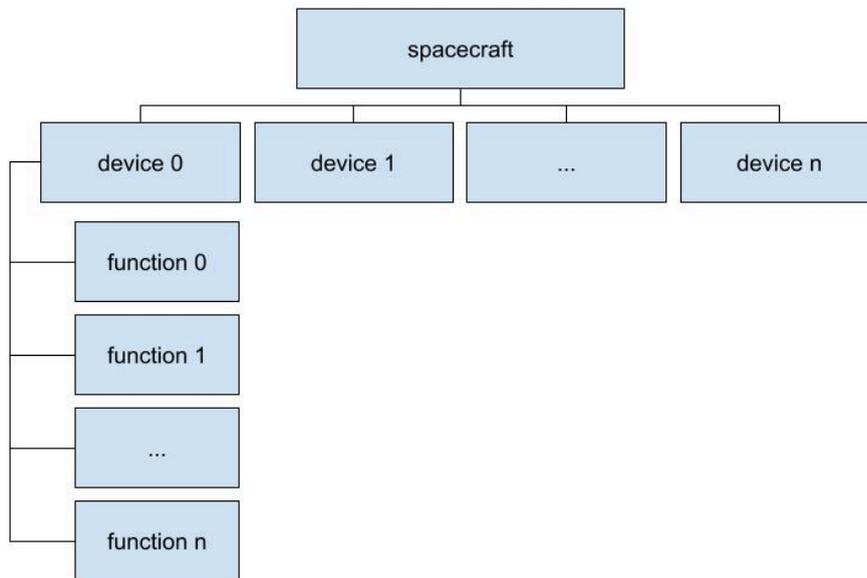
В сервисах семейства "Orbita" космический аппарат представлен глобальным объектом класса "Spacecraft", который хранит все параметры аппарата, его бортовые устройства, а также имеет некоторые типовые функции, такие как отсчет времени от момента старта миссии.

В частности, в состав аппарата и/или наземной станции входит набор бортовых устройств, представленных объектами класса "Device", перечисленных в неизменяемом массиве "Spacecraft.Devices".

У каждого устройства в аппарате есть порядковый номер (начинается с 0). Кроме того, у него может быть уникальное название, задаваемое при конструировании аппарата. Обращение к устройству обычно осуществляется через этот порядковый номер, например:

```
accumulator = spacecraft.devices[0];
```

Каждое устройство может выполнять одну или несколько функций, определяющих его роль в аппарате (например, аккумулятор или радиопередатчик) и представленных объектами класса Function и классов-наследников (Accumulator, Heater, ...). У каждой функции в устройстве также есть порядковый номер (также начинается с 0) и может быть название.



На старте миссии конструкция аппарата считается заданной.

Пример массива устройств аппарата

Номер	Тип устройства	Включено в начале симуляции	Функции устройства и их порядковые номера	Примечание
0	Аккумулятор	True	Аккумулятор (0)	Может быть заряжен изначально, накапливает заряд при избытке питания и отдает при недостатке

1	Система солнечных батарей	True	Солнечная панель (0), Солнечная панель (1), Солнечная панель (2), Солнечная панель (3), Солнечная панель (4), Солнечная панель (5)	Набор из нескольких солнечных панелей, ориентированных на 6 разных гранях аппарата, обеспечивают питание аппарата, если на них падает солнечный свет
2	Маневровый двигатель	False	thruster (0), propellant-tank(1)	Небольшой бортовой реактивный двигатель для смены орбиты. Содержит реактивное устройство и "топливный" бак (например холодный газ под давлением)

Пример устройства и значение его функций в начальный момент симуляции

Устройство: маневровый двигатель

Номер функции	Тип функции	Включена в начале	Значение при обращении	Комментарий
0	Реактивный двигатель	False	0	Функция устанавливает и возвращает тягу. На начало симуляции двигатель доступен, но вывод рабочего тела не осуществляется.
1	Топливный бак	True	0.572	Запас рабочего тела

3.2. Написание программы управления на языке JavaScript

Ниже приведена краткая справка по написанию программы на языке JavaScript, достаточная для написания простого кода управления аппаратом.

Общие положения

Первой строкой программы является директива, позволяющая использовать более строгий вариант JavaScript:

```
'use strict';
```

После каждой инструкции (команды) должна ставиться точка с запятой. Переменные объявляются с помощью ключевых слов `var` или `let`. Чтобы переменную можно было использовать во всех функциях скрипта, можно сделать её глобальной: в таком случае её необходимо объявить в начале скрипта, до всех функций.

```
var thruster;
```

Инициализировать переменную, то есть, сохранить в неё какое-то конкретное значение, можно с помощью операции присваивания - знака равенства (не путать с двойным знаком равенства - этот оператор позволяет сравнивать выражения).

```
let time = 0;
```

Для условного ветвления существует инструкция `if`, к которой можно также добавить блоки `else if` и `else`. Проверяемые условия записываются в круглых скобках после слова `if`. Если выражение, записанное в условии, истинное, то выполняется соответствующий блок.

Инструкции, которые должны быть выполнены, берутся в фигурные скобки.

```
if (time == 10) {  
  // делать что-то  
} else if (time > 15) {  
  // делать что-то ещё  
} else {  
  // делать что-то совсем другое  
}
```

Доступ к свойствам объекта осуществляется через операцию точка, которая ставится после имени интересующего объекта.

Обратимся к свойствам класса `Spacecraft`: если мы хотим узнать время, прошедшее с начала полёта, необходимо сначала записать имя конкретного объекта - `spacecraft` (это спутник, для которого вы пишете программу), - затем точку, после чего обратиться к соответствующему свойству - `flight_time`. Можно сразу сохранить значение времени в объявленную ранее переменную `time`:

```
time = spacecraft.flight_time;
```

Класс `Spacecraft` содержит в себе несколько объектов класса `Device`, имя которых записывается как `devices[i]`, где `i` - номер устройства в таблице, данной выше. Внутри каждого объекта класса `Device` так же есть один или несколько объектов класса `Function`, к которым можно обратиться по имени `functions[j]`, где `j` - номер функции в таблице. Для каждого объекта класса `Device` существуют свойства, приведённые в списке выше. Допустим, мы хотим узнать угловую скорость по направлению `Y` с помощью гироскопа и сохранить её в некоторую переменную `Vy`.

Список возможного оборудования и его функций можно посмотреть в следующем разделе.

Состав устройств конкретного аппарата, порядок его устройств и функций или задан в условиях сценария, или задается пользователем самостоятельно в режиме свободного проектирования.

Допустим, среди бортовых устройств есть Гироскоп (его номер 2) и среди его функций есть датчик угловой скорости по оси X (его номер 0).

Для получения значения угловой скорости в текущий момент обращаемся сперва к аппарату - `spacescraft`, затем к устройству - `devices[2]`, затем к функции - `functions[0]`, и в последнюю очередь - к самому свойству для функции гироскопа - `angular_velocity`.

```
var Vy = spacecraft.devices[2].functions[0].angular_velocity;
```

Чтобы не писать каждый раз такие громоздкие конструкции, можно создавать специальные переменные (лучше всего, с говорящим названием) и затем обращаться к требуемым свойствам через них.

Например, чтобы изменить тягу двигателя (свойство `thrust` класса "Двигатель" функцией "Управление тягой"), можно сначала определить переменную `thruster`, которая сразу позволит обращаться к необходимой функции, а затем уже менять свойство `thrust`.

```
var thruster = spacecraft.devices[0].functions[0];
```

После чего зададим двигателю максимальную тягу с помощью свойства `maximum_thrust`:

```
var thruster = spacecraft.devices[0].functions[0];  
thruster.thrust = thruster.maximum_thrust;
```

Структура программы управления

Глобальные переменные и директивы объявляются в начале программы. Если необходимо выполнить какой-либо код один раз в начале симуляции, следует поместить его в функцию `setup()`.

Основной код управления помещается внутри функции `loop()`, которая вызывается циклически все время симуляции. Во избежание прерывания функции по таймеру с ошибкой следует избегать длительного исполнения кода в этой функции.

Ниже приведен пример простой программы, где для удобства создается переменная для обращения к функции "Управление тягой" объекта класса "Двигатель". Двигатель включен в начале симуляции, но начиная с сотой миллисекунды полета должна быть задана тяга, равная нулю, - то есть, двигатель можно считать выключенным.

```

'use strict'; // Объявлена директива
var thruster; // Объявлена переменная для удобства работы с устройством
"Двигатель"

// В начале симуляции устанавливаем, что "thruster" указывает на первую
функцию первого устройства.
function setup() {
  thruster = spacecraft.devices[0].functions[0];
}

// Отключаем тягу начиная с сотой миллисекунды
function loop() {
  if (spacecraft.flight_time >= 100) {
    thruster.thrust = 0;
  }
}

```

3.3. Обзор устройств и функций космического аппарата

Ниже приведены свойства и методы, которые необходимо использовать при написании бортовой программы управления космическим аппаратом.

Напоминаем, что минимальная управляемая единица оборудования - это функция. При создании аппарата вы можете добавить к любым устройствам любые функции. Например, вы можете оснастить спутник средствами связи. Для этого можно:

1. Добавить устройство "Радиомодуль" с тремя функциями - "Приемник", "Передатчик" и "Антенна", или
2. Сделать два разных устройства с одной функцией каждое, одно - только для приема, другое только для передачи, и отдельно третье - антенну;
3. Сделать устройство с несколькими функциями "Передатчик", если оно может независимо работать на нескольких разных частотах
4. Добавить в устройство "Научный модуль" функцию "Передатчик", если по задумке разработчика это оборудование имеет свой радиомодуль и антенну.
5. И т.п.

Свойства класса Spacecraft

Свойство	Значение
flight_time	Значение времени от начала полета в секундах (с точностью до миллисекунд).

calendar_utc_time	Глобальное время в симуляции по UTC
-------------------	-------------------------------------

devices	Массив, содержащий устройства (объекты класса Device).
---------	--

Общие свойства класса Device

Свойство	Значение
enabled	Состояние устройства (true - включено, false - выключено)
failed	Признак того, что устройство вышло из строя (true - вышло из строя, false - работоспособно).
functions	Массив, содержащий функции (объекты класса Function и классов-наследников)

Пример использования:

```
solar_panel_x_bottom = spacecraft.devices[0].function[1]; // устанавливаем, что переменная solar_panel_x_bottom - это вторая функция первого устройства спутника
```

Общие методы класса Device

Метод	Описание
enable()	Включить устройство
disable()	Выключить устройство

Пример использования:

```
spacecraft.devices[1].disable(); // Выключаем второе устройство в списке
```

Обзор конкретных аппаратных функций приведен в разделах ниже

3.3.1. Аккумулятор

Функция, отражающая бортовую аккумуляторную батарею. Если она и содержащее ее устройство включены:

1. Батарея заряжается при избытке поступающей от солнечных панелей или иных источников питания;
2. Батарея отдает свой заряд при недостатке поступающей от солнечных панелей или иных источников питания.

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
charge	Текущий оставшийся заряд в Джоулях
capacity	Емкость батареи в Джоулях
empty	Батарея полностью разряжена (true/false)
full	Батарея полностью заряжена (true/false)

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

Пример использования:

```
accumulator = spacecraft.devices[0].functions[0];
```

```
if (accumulator.charge > 1000) {  
  // Включить что-то, что могло быть выключено из-за недостатка питания
```

3.3.2. Солнечная панель

Функция, отражающая бортовую солнечную панель. Если она и содержащее ее устройство включены:

1. Обеспечивается поступление питания в бортовую систему электропитания, в зависимости от освещенности панели;

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

3.3.3. Маховик

Функция, отражающая бортовой двигатель ориентации на основе механического маховика. Если она и содержащее ее устройство включены:

1. Маховик вращается с установленной скоростью, что отражается на ориентации и скорости вращения аппарата;

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).

motor_torque	Крутящий момент мотора маховика. Важно: у мотора есть ограничение в максимальный момент см. maximum_motor_torque.
maximum_motor_torque	Максимальный момент маховика.
inertia_moment	Момент инерции маховика.
angular_velocity	Угловая скорость вращения маховика.

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

3.3.4. Датчик угловой скорости

Функция, отражающая бортовой датчик угловой скорости (гироскоп). Если она и содержащее ее устройство включены:

1. Можно получать значение угловой скорости по одной оси;

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
angular_velocity	Угловая скорость космического аппарата по одной из осей.

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

3.3.5. Навигационный датчик

Функция, отражающая бортовой датчик положения в глобальной системе навигации (GPS, ГЛОНАСС или иных). Если она и содержащее ее устройство включены:

1. Можно получать значение текущих координат аппарата в пространстве относительно Земли (широта, долгота, высота);

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
location	Массив, содержащий координаты аппарата относительно поверхности Земли (широта в градусах, долгота в градусах, высота над уровнем моря в метрах).

Важно: максимальная высота, при которой работает навигатор, зависит от характеристик конкретного устройства!

Методы функции

Метод	Описание
enable()	Включить функцию

disable()	Выключить функцию
-----------	-------------------

3.3.6. Звездный датчик

Функция, отражающая бортовой датчик ориентации аппарата относительно звезд. Если она и содержащее ее устройство включены:

1. Можно получать значение абсолютной ориентации аппарата;

Свойства функции

Свойство	Описание
----------	----------

enabled	Состояние функции устройства (true - включена, false - выключена)
---------	---

failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
--------	---

orientation	Массив из 4 чисел представляющих кватернион положения аппарата относительно звёзд. Скалярная часть идёт в конце.
-------------	--

Методы функции

Метод	Описание
-------	----------

enable()	Включить функцию
----------	------------------

disable()	Выключить функцию
-----------	-------------------

3.3.7. Реактивный двигатель

Функция, отражающая бортовой маневровый реактивный двигатель ориентации на основе расхода рабочего тела, например сжатого холодного газа. Если она и содержащее ее устройство включены:

1. Выпускает рабочее тело потоком указанной тяги;

2. ВАЖНО: В составе аппарата должна быть хотя бы одна функция "топливный бак" и остаток рабочего тела должен быть достаточным для поддержания указанной тяги. Тяга должна быть в пределах технических ограничений устройства.

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
maximum_thrust	Максимально возможная тяга (только чтение)
minimum_thrust	Минимально возможная тяга (только чтение)
thrust	Установленная тяга (можно задавать значение).

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

3.3.8. Топливный бак

Функция, отражающая топливный бак для бортового маневрового реактивного двигателя ориентации на основе расхода рабочего тела, например сжатого холодного газа. Если она и содержащее ее устройство включены:

1. Предоставляет топливо для реактивного двигателя;

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
amount	Текущий остаток топлива
capacity	Емкость топливного бака

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

3.3.9. Передатчик

Функция, отражающая бортовой радио передатчик. Если она и содержащее ее устройство включены:

1. Можно отправлять данные для передатчиков в прямой видимости (если не указано иначе);

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)

failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
--------	---

Методы функции

Метод	Описание
-------	----------

enable()	Включить функцию
----------	------------------

disable()	Выключить функцию
-----------	-------------------

transmit(a)	Отправляет буфер a типа Uint8Array в очередь на отправку, которая будет отправляться с доступной передатчику скоростью в эфир
-------------	---

3.3.10. Приемник

Функция, отражающая бортовой радио приёмник. Если она и содержащее ее устройство включены:

1. Можно получать данные от передатчиков в прямой видимости (если не указано иначе);

Свойства функции

Свойство	Описание
----------	----------

enabled	Состояние функции устройства (true - включена, false - выключена)
---------	---

failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
--------	---

Методы функции

Метод	Описание
-------	----------

enable()	Включить функцию
----------	------------------

disable()	Выключить функцию
-----------	-------------------

receive()	Возвращает байтовый массив типа Uint8Array, данных которые успели попасть в буфер приёма, после вызова буфер очищается. Может принимать параметр число - количество байт для чтения из буфера. В зависимости от ситуации может прийти не всё сообщение (см. ниже подробности).
-----------	--

Важные замечания по приему:

1. Принимается отправленные другим устройством типа Передатчик данные в том объеме, в котором они могут быть приняты с учетом законов физики, с учетом видимости, частоты и скорости передачи.
2. Сообщение приходит байтами. Сообщение может прийти не целиком, если связь прервалась, или не все байты успели быть отправлены.

3.3.11. Источник данных

Функция, отражающая бортовую полезную нагрузку. Если она и содержащее ее устройство включены:

1. Можно получать данные от полезной нагрузки, в зависимости от задачи они могут различаться;

Свойства функции

Свойство	Описание
----------	----------

enabled	Состояние функции устройства (true - включена, false - выключена)
---------	---

failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
--------	---

Методы функции

Метод	Описание
-------	----------

enable()	Включить функцию
----------	------------------

disable()	Выключить функцию
-----------	-------------------

read()	Возвращает байтовый массив типа Uint8Array, данных которые успели накопиться в буфере, после вызова буфер очищается. Стоит обратиться к документации JS за особенностями работы с Uint8Array. Может принимать параметр число - количество байт для чтения из буфера
--------	---

3.3.12. Потребитель данных

Функция, отражающая анализ полученной полезной информации. Может располагаться в том числе на наземных станциях приема данных. Если она и содержащее ее устройство включены:

1. Можно передавать данные полезной нагрузке, в зависимости от задачи они могут различаться. Важно соблюдать формат сообщения.;

Свойства функции

Свойство	Описание
----------	----------

enabled	Состояние функции устройства (true - включена, false - выключена)
---------	---

failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).
--------	---

Методы функции

Метод	Описание
-------	----------

enable()	Включить функцию
----------	------------------

disable()	Выключить функцию
-----------	-------------------

write(a)	Отправляет буфер a типа Uint8Array в очередь на отправку. Стоит обратиться к документации JS за особенностями работы с Uint8Array.
----------	--

3.3.13. Нагреватель

Функция, отражающая бортовой нагреватель. Если она и содержащее ее устройство включены:

1. Можно увеличить рассеяние мощности в аппарате, за счёт потребления электрической мощности;

Свойства функции

Свойство	Описание
enabled	Состояние функции устройства (true - включена, false - выключена)
failed	Признак того, что функция вышла из строя (true - вышла из строя, false - работоспособна).

Методы функции

Метод	Описание
enable()	Включить функцию
disable()	Выключить функцию

4. Активация лицензии (только для администраторов)

Для получения статуса администратора лицензии зарегистрированному пользователю необходимо кликнуть на "Профиль" в правом верхнем углу сайта, затем найти строку "добавить лицензию", ввести в появившееся поле код лицензии, поставить галочку в поле "I'm not a robot" и кликнуть на "Получить доступ".

Информация по лицензиям:

Лицензия 1 (неограниченная)

[добавить лицензию](#)

Код лицензии

XXXX



XXXX



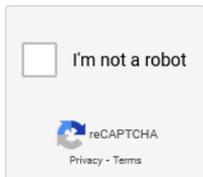
XXXX



XXXX



Обязательное поле!



Получить доступ

По управлению лицензией и администрированию событий смотри Руководство администратора.