

API уровня подсистем аппарата

Методы и результаты

Обращение к параметрам и функциям аппаратов осуществляется через вызов методов. Вызов методов каждой из подсистем возвращает запрашиваемый результат или None.

Ошибки

Ошибки возвращаются в виде исключений. Возможные классы исключений:

`GenericError`

Базовый класс для ошибок в программе.

`SystemNotAvailableError`

Обращение к системе, которая отсутствует в аппарате

`NotSupportedError`

Обращение к функции, которая не поддерживается аппаратом

`BadParametersError`

При вызове метода были переданы недопустимые значения аргументов.

Константы (Состояния подсистем аппарата)

`STATE_OFF`

Устройство выключено.

`STATE_ON`

Устройство включено.

`STATE_SLEEP`

Устройство находится в режиме экономии энергии (сна).

`STATE_DEAD`

Устройство неисправно.

Подсистемы

Управление на уровне подсистем осуществляется через объект класса Sputnik.

Космический аппарат состоит из подсистем, которые представлены как объекты классов подсистем:

1. sputnik.cpu - БЦВМ (класс CPU);
2. sputnik.telemetry - подсистема телеметрии (класс Telemetry);
3. sputnik.transmitter - подсистема высокопроизводительной связи (класс Transmitter);
4. sputnik.power - подсистема электропитания (класс Power);
5. sputnik.navigation - подсистема навигации (класс Navigation);
6. sputnik.orientation - подсистема управления ориентацией и стабилизацией (класс Orientation);
7. sputnik.engine - подсистема изменения орбиты (класс Engine);
8. sputnik.heat_control - подсистема обеспечения теплового режима (класс HeatControl);
9. sputnik.camera - подсистема полезной нагрузки (камеры) (класс Camera);
10. sputnik.container - подсистема полезной нагрузки (контейнера) (класс Container).

Класс CPU

Подсистема БЦВМ.

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы БЦВМ. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state БЦВМ. Не возвращает данных.

```
get_flight_time(без параметров)
```

Получить время в с с момента включения БЦВМ в начале полета. Возвращает время в секундах как дробное число.

Класс Telemetry

Подсистема телеметрии.

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы телеметрии. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state телеметрии. Не возвращает данных.

```
send_message (msg)
```

Отправить строку msg через радио-канал телеметрии (для получения на наземных измерительных пунктах). Не возвращает данных.

Класс Power

Подсистема электропитания.

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы электропитания. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state подсистеме электропитания. Не возвращает данных.

```
get_battery_capacity()
```

Получить значение текущей емкости аккумулятора (Вт ч).

```
get_generation()
```

Получить мощность генерируемого тока (Вт).

```
get_consumption()
```

Получить мощность потребляемого аппаратом тока (Вт).

Класс Navigation

Подсистема навигации.

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы навигации. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state подсистеме навигации. Не возвращает данных.

```
get_orbit_height()
```

Возвращает текущую высоту орбиты аппарата (м).

```
get_z_axis_angle()
```

Возвращает угол положения аппарата относительно оси Z (градусы).

```
get_x_coord()
```

Возвращает координату положения аппарата по оси X (м).

```
get_y_coord()
```

Возвращает координату положения аппарата по оси Y (м).

```
get_transversal_velocity()
```

Получить трансверсальную составляющую скорости аппарата (м/с).

```
get_radial_velocity()
```

Получить радиальную составляющую скорости аппарата (м/с).

Класс Orientation

Подсистема управления ориентацией и стабилизацией.

Некоторые методы этого класса в качестве первого аргумента принимают константу оси, вокруг которой осуществляется вращение. Ось задается одной из следующих констант:

- `AXIS_X` - ось X;
- `AXIS_Y` - ось Y;
- `AXIS_Z` - ось Z.

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы управления ориентацией и стабилизацией. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state подсистеме управления ориентацией и стабилизацией. Не возвращает данных.

```
get_angle(axis)
```

Получить угол ориентации аппарата относительно оси axis. Возвращается значение в градусах.

```
get_angular_velocity(axis)
```

Получить скорость вращения аппарата относительно оси axis. Возвращается значение в градусах/с.

```
start_motor(axis):
```

Включить маховик, который осуществляет вращение вокруг оси axis. Не возвращает данных.

```
stop_motor(axis)
```

Выключить маховик, который осуществляет вращение вокруг оси axis. Не возвращает данных.

```
set_motor_moment(axis, torsion)
```

Задать момент вращения torsion маховику, который осуществляет вращение вокруг оси axis. Может быть вызвана только когда мотор включен. Не возвращает данных.

```
start_coil(axis)
```

Включить катушку, которая осуществляет стабилизацию аппарата вокруг оси axis. Не возвращает данных.

```
stop_coil(axis)
```

Выключить катушку, которая осуществляет стабилизацию аппарата вокруг оси axis. Не возвращает данных.

Класс Engine

Подсистема изменения орбиты (двигатель).

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы навигации. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state подсистеме навигации. Не возвращает данных.

```
get_fuel()
```

Получить объем доступного топлива в топливных баках (кг).

```
start_engine()
```

Включить двигатель. Не возвращает данных.

```
stop_engine()
```

Выключить двигатель. Не возвращает данных.

```
set_traction(t)
```

Установить массовый расход топлива t в кг/с. Может быть вызвана только когда двигатель включен. Не возвращает данных.

Класс HeatControl

Подсистема обеспечения теплового режима аппарата.

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы обеспечения теплового режима аппарата. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние $state$ подсистеме обеспечения теплового режима аппарата. Не возвращает данных.

```
get_temperature()
```

Получить текущую температуру внутри аппарата, К.

```
start_heating()
```

Включить нагреватель. Не возвращает данных.

```
stop_heating()
```

Выключить нагреватель. Не возвращает данных.

```
set_power(p)
```

Установить мощность нагревателя p (Вт). Может быть вызвана только при включенном нагревателе. Не возвращает данных.

Класс Camera

Подсистема полезной нагрузки (камеры).

Методы:

```
get_state (без параметров)
```

Получить текущее состояние камеры. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state камере. Не возвращает данных.

```
take_photo()
```

Сделать мгновенный снимок камерой. Возвращается номер блока памяти, в который был помещен снимок.

```
start_shooting()
```

Начать съемку камерой. Поток данных записывается в память. Метод не возвращает данных.

```
stop_shooting()
```

Закончить съемку камерой. Возвращается номер блока памяти, в который были помещены данные.

```
get_image_size(slot_num)
```

Получить размер снимка (или потока видео), расположенного в блоке памяти с номером slot_num. Возвращает число байт.

Класс Transmitter

Подсистема полезной нагрузки (высокопроизводительной связи).

Методы:

```
get_state (без параметров)
```

Получить текущее состояние подсистемы высокопроизводительной связи. Возвращает состояние (см. выше).

```
set_state (state)
```

Задать новое состояние state подсистеме высокопроизводительной связи. Не возвращает данных.

```
send_data (msg_type, data, receiver, sender, timeout)
```

Отправить блок данных data типа msg_type на наземный измерительный пункт receiver (номер) от передатчика sender (номер) со сроком передачи timeout (с). Необязательными параметрами являются:

- receiver - тогда сообщение передается для всех наземных пунктов (receiver == -1);
- sender - тогда считается, что источником сообщения является сам аппарат (sender == 0);
- timeout - тогда считается, что сообщение имеет бесконечное время доставки.

Могут быт переданы сообщения следующих типов:

- MESSAGE_PHOTO - сообщение со снимком камеры аппарата;
- MESSAGE_SMS - сообщение от системы обмена короткими сообщениями;
- MESSAGE_TELEMETRY - сообщение с телеметрией.

Не возвращает данных.

```
send_photo(slot_num, receiver)
```

Отправить на Землю снимок, который находится в блоке памяти под номером slot_num на наземный измерительный пункт receiver. Второй параметр не является обязательным: тогда сообщение передается для всех наземных пунктов (receiver == -1). Не возвращает данных.

```
receive(sender)
```

Инициировать получение сообщения от передатчика с номером sender. Не возвращает данных.

```
get_progress(sender)
```

Получить прогресс получения сообщения от наземного передатчика с номером sender. Возвращает проценты (от 0 до 100) как дробное число.

```
get_message(sender)
```

Получить сообщение от наземного передатчика с номером sender. Возвращает структуру с сообщением, если сообщение целиком получено аппаратом (предыдущий вызов get_progress вернул 100).

Класс Container

Подсистема полезной нагрузки (контейнер для проведения научных экспериментов).

Методы:

```
get_state (без параметров)
```

Получить текущее состояние контейнера. Возвращает состояние (см. выше).


```
set_state (state)
```

Задать новое состояние state контейнеру. Не возвращает данных.

```
start_experiment()
```

Начать научный эксперимент. Не возвращает данных.

```
stop_experiment()
```

Закончить научный эксперимент. Не возвращает данных.

```
set_parachute_height(h)
```

Установить раскрытие парашюта контейнера при достижении высоты h (м).

```
drop()
```

Сбросить контейнер. В некоторых конструкциях это приводит к разрушению самого аппарата. Не возвращает данных.

From:

<http://wiki.orbicraft.ru/> - **ОрбиКрафт: Космическая инженерия для начинающих**

Permanent link:

<http://wiki.orbicraft.ru/doku.php?id=systems-api>

Last update: **2016/03/07 14:19**

