

# Простое API (для миссий посадки на планеты)

Здесь приводится пример API на языке Python, поскольку интерфейс реализован только на этом языке.

## Методы и результаты

Обращение к параметрам и функциям аппаратов осуществляется через вызов методов. Каждый вызов возвращает 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.

## Состояние аппарата (константы)

```
STATE_OFF
```

Устройство выключено.

```
STATE_ON
```

Устройство включено.

```
STATE_DEAD
```

Устройство неисправно.

## Класс Probe

Управление аппаратом осуществляется через объект **probe** класса Probe.

Методы:

```
run()
```

Системный метод, который должен вызываться на каждом цикле исполнения программы, например:

```
while probe.run():  
    # делать что-то
```

```
get_device_state(device_id)
```

Получить текущее состояние устройства с кодом device\_id. Возвращает состояние (см. выше).

```
set_device_state(device_id, state)
```

Задать новое состояние state устройству с кодом device\_id. Не возвращает данных.

```
get_device_period(device_id)
```

Получить значение периода генерации данных/передачи для устройства с кодом device\_id. Возвращает дробное число в секундах.

```
set_device_period(device_id, period)
```

Задать новый период period устройству с кодом device\_id. Не возвращает данных.

```
cpu_get_flight_time()
```

Получить время в секундах с момента начала миссии. Возвращает время в секундах как дробное число.

```
navigation_get_accel()
```

Получить значение полного ускорения аппарата. Возвращает ускорение в м/кв.с как дробное число.

```
navigation_has_landed()
```

Проверить, совершил ли аппарат посадку. Возвращает True, если аппарат совершил посадку.

```
power_get_battery_capacity()
```

Получить значение текущей запасенной энергии аккумулятора, Вт ч.

```
power_get_generation()
```

Получить мощность генерируемого тока (Вт).

```
power_get_consumption()
```

Получить мощность потребляемого аппаратом тока (Вт).

```
transmitter_get_bandwidth()
```

Получить ширину доступного канала передачи данных, Кбит/с.

```
transmitter_get_traffic()
```

Получить объем генерируемого устройствами трафика, Кбит/с.

```
heat_control_get_ext_temperature()
```

Получить текущую температуру снаружи аппарата. Возвращает температуру в К как дробное число.

```
telemetry_send_message(msg)
```

Отправить строку `msg` в канал телеметрии (для дальнейшего получения его на Земле). Не возвращает данных. **Внимание:** эта функция доступна только если аппарат содержит включённое устройство расширенной телеметрии.

```
engine_get_fuel()
```

Получить массу доступного топлива в топливных баках. Возвращает массу топлива в кг.

```
engine_set_angle(device_id, angle)
```

Задать угол наклона `angle` (в градусах) двигателя с кодом `device_id`. Не возвращает данных.

From:

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

Permanent link:

<http://wiki.orbcraft.ru/doku.php?id=probe-api&rev=1481835856>



Last update: **2016/12/15 21:04**