



Всероссийский конкурс  
«КАДРЫ ДЛЯ ЦИФРОВОЙ ПРОМЫШЛЕННОСТИ. СОЗДАНИЕ ЗАКОНЧЕННЫХ  
ПРОЕКТНО-КОНСТРУКТОРСКИХ РЕШЕНИЙ В РЕЖИМЕ СОРЕВНОВАНИЙ  
«КИБЕРДРОМ»



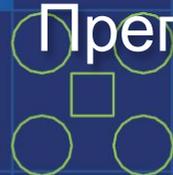
**Кибердром** 2022

ГК «Геоскан»

Модуль: ВВЕДЕНИЕ В ДЕЯТЕЛЬНОСТЬ  
«ЭКСПЛУАТАЦИЯ БЕСПИЛОТНЫХ АВИАЦИОННЫХ  
СИСТЕМ»

Название темы: Введение в программное обеспечение  
конкурсного задания

Преподаватель: Самарин Алексей Сергеевич, инженер-  
программист



Москва, 2022





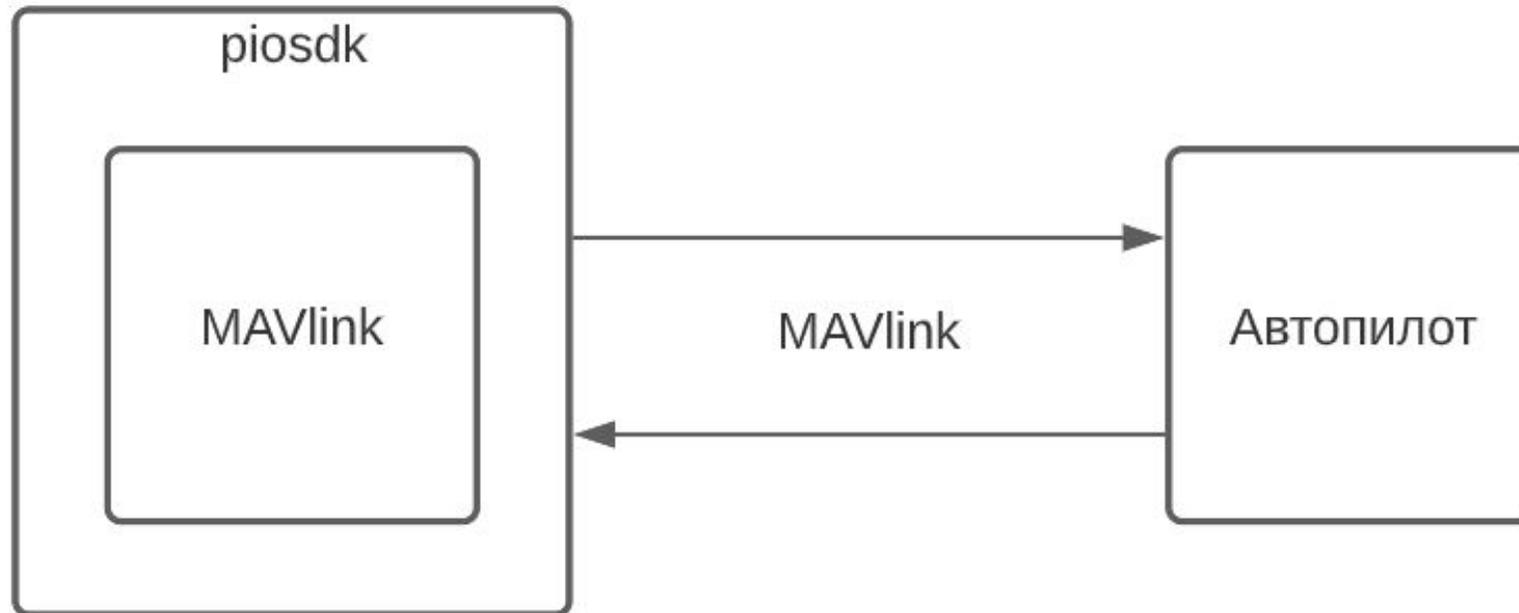
- Что такое piosdk
- Установка piosdk
- Работа с piosdk
- Установка симулятора
- Работа с симулятором



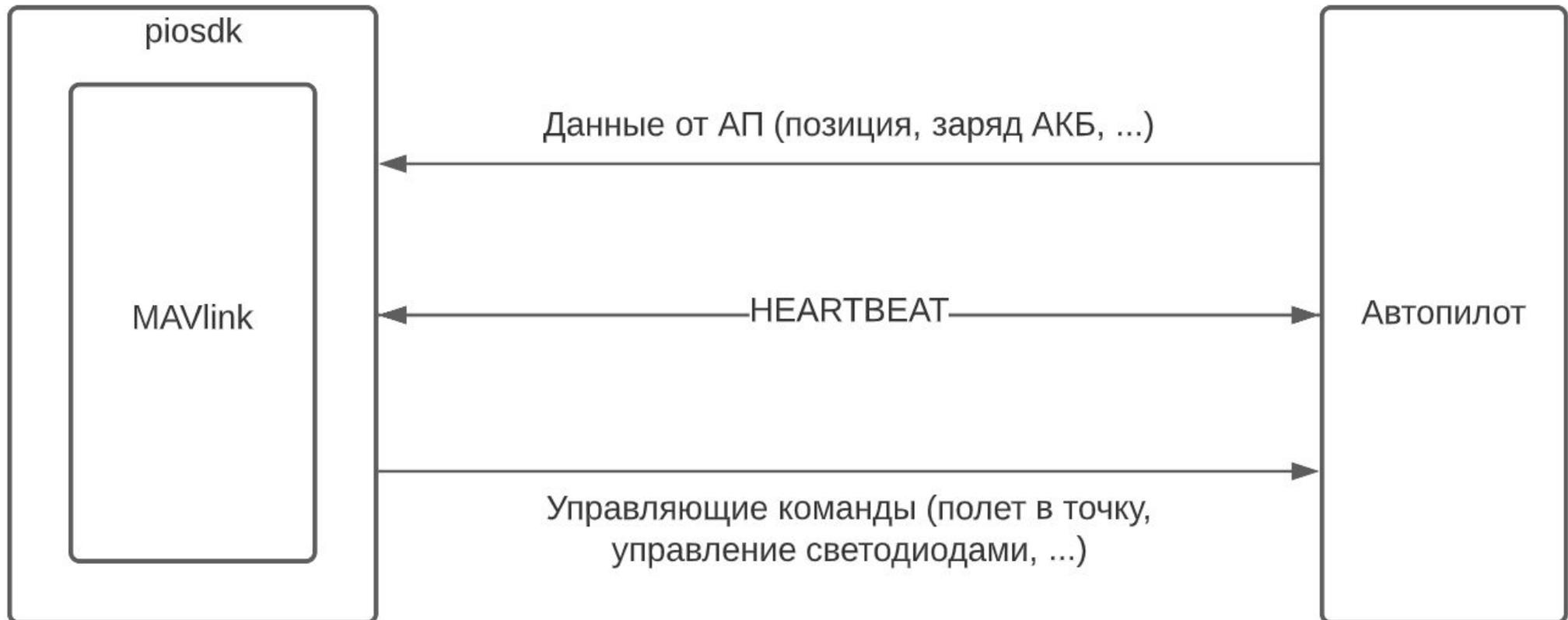
- получение знаний и опыта подготовки полетных заданий с использованием piosdk;
- запуск полетных заданий в симуляторе.



Полную версию занятия можно найти по ссылке:  
[https://docs.google.com/document/d/1yprM6u8ZKvVlpqAhEHG9g8NeQ7mmD1\\_SRB383B2fdlw/edit#](https://docs.google.com/document/d/1yprM6u8ZKvVlpqAhEHG9g8NeQ7mmD1_SRB383B2fdlw/edit#)



Ссылка на протокол MAVlink: <https://mavlink.io/en/>



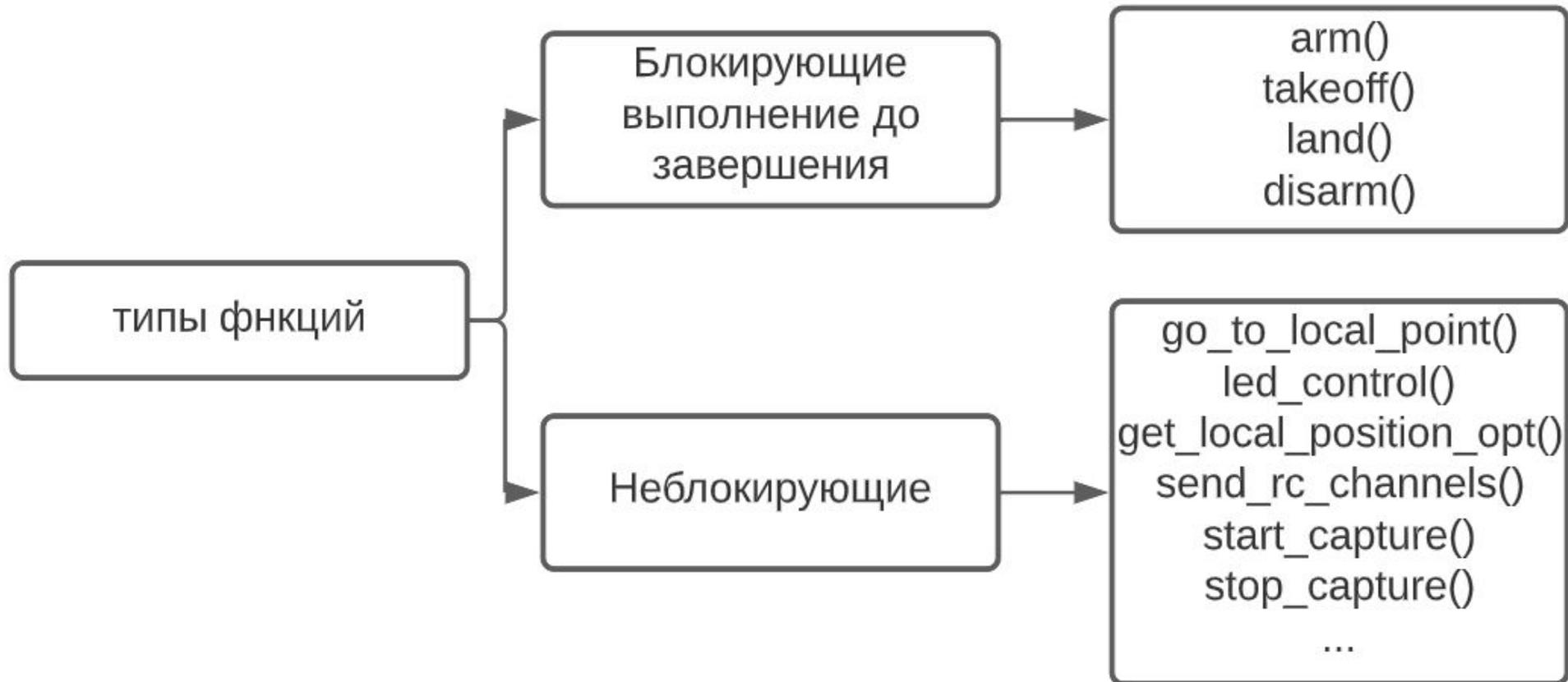
Ссылка на протокол MAVlink: <https://mavlink.io/en/>



```
1) import Pioneer
```

```
2) drone = Pioneer()
```

```
3) drone.arm()
```



Название поля	Описание
<b>method</b>	Метод соединения: <ul style="list-style-type: none"><li>● 0 - соединение по wifi (pioneer mini, pioneer + gs_router)</li><li>● 1 - соединение по uart</li><li>● 2 - соединение по wifi (pioneer + mavrouter)</li></ul> * По умолчанию 0
<b>pioneer_ip</b>	ip используемого пионера
<b>pioneer_mavlink_port</b>	port используемого пионера
<b>device</b>	устройство uart (при method = 1)
<b>baud</b>	скорость uart (при method = 1)
<b>logger</b>	вывод отладочной информации в терминал * по умолчанию True

```
1 from piosdk.piosdk import Pioneer
2
3 drones = []
4
5 drones.append(Pioneer(method=2, pioneer_ip="127.0.0.1", pioneer_mavlink_port=8000, logger=True))
6 drones.append(Pioneer(method=2, pioneer_ip="127.0.0.1", pioneer_mavlink_port=8001, logger=False))
7 drones.append(Pioneer(method=2, pioneer_ip="127.0.0.1", pioneer_mavlink_port=8002, logger=False))
8 drones.append(Pioneer(method=2, pioneer_ip="127.0.0.1", pioneer_mavlink_port=8003, logger=False))
```

В примере выше показано создание 4х экземпляров класса Pioneer, с добавлением их в список drones. (добавлять в список совсем не обязательно)

Все экземпляры созданы на локальном ip = 127.0.0.1, но каждый имеет разный порт. В drones[0] параметр logger = True, что приведет к выводу лога в терминал.

## Обзор функций. arm, disarm, takeoff, land

```
1 import time
2 from piosdk.piosdk import Pioneer
3
4 drone = Pioneer(method=2, pioneer_ip="127.0.0.1", pioneer_mavlink_port=8000, logger=True)
5
6 drone.arm()
7 drone.takeoff()
8 time.sleep(2)
9 drone.land()
```

arm() - завести моторы

disarm() - остановить моторы

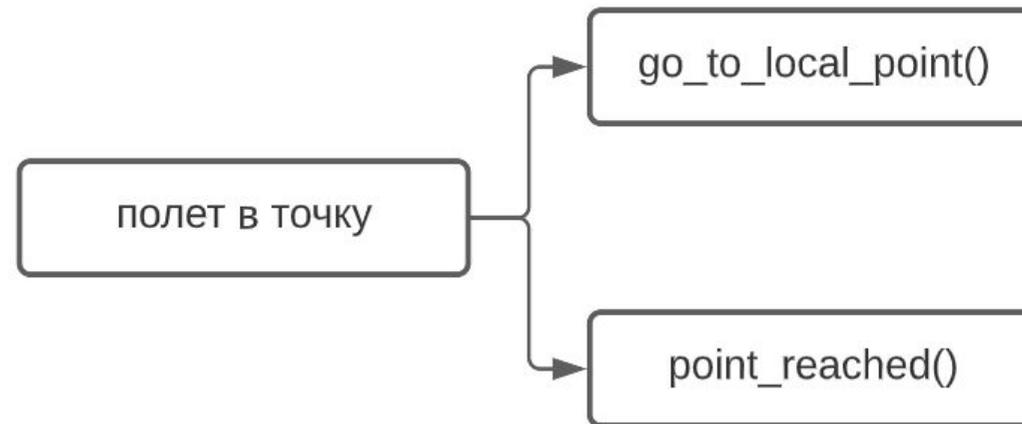
takeoff() - взлететь

land() - выполнить посадку

Все функции не имеют никаких параметров

### \* Важно:

1. Вызов функции takeoff() должен происходить только после функции arm()
2. Вызов функции disarm() во время полета приведет к остановке моторов и падению квадрокоптера.



Параметры функции `go_to_local_point()`

Название поля	Описание
<b>x</b>	Координата X
<b>y</b>	Координата Y
<b>z</b>	Координата Z
<b>yaw</b>	Угол поворота по yaw

```
1  import Pioneer
2  import time
3
4  drone = Pioneer()
5
6  drone.arm()
7  drone.takeoff()
8
9  drone.go_to_local_point(x=1, y=1, z=1)
10 time.sleep(5)
11 drone.go_to_local_point(x=5, y=1, z=1)
12 time.sleep(5)
13 drone.go_to_local_point(x=1, y=1, z=1)
14 time.sleep(5)
15 drone.go_to_local_point(x=5, y=1, z=1)
16 time.sleep(5)
17 drone.land()
```

go\_to\_local\_point() + point\_reached(). Пример  
использования.

```
4 pioneer = Pioneer(method=2, pioneer_ip='10.10.0.88', pioneer_mavlink_port=5656)
5
6 # Точки полетного задания []
7 mission_point = [[1.5, 1.5, 2],
8                  [1.5, 7, 2],
9                  [7, 7, 2],
10                 [7, 1.5, 2],
11                 [1.5, 1.5, 2]]
12
13 # Индекс текущей точки, куда нужно будет лететь
14 current_mission_point = 0
15
16 if __name__ == '__main__':
17     pioneer.arm()
18     pioneer.takeoff()
19     time.sleep(2)
20
21     new_point = True # флаг того, что нужно выполнить go_to_local_point
22     while True:
23         if new_point:
24             pioneer.go_to_local_point(x=mission_point[current_mission_point][0],
25                                     y=mission_point[current_mission_point][1],
26                                     z=mission_point[current_mission_point][2],
27                                     yaw=0)
28
29             new_point = False
30
31         # Проверка долетел ли Пионер до точки
32         if pioneer.point_reached(True):
33             # выставляем флаг того, что нужно выполнить go_to_local_point
34             # и увеличиваем индекс точки куда лететь
35             new_point = True
36             current_mission_point += 1
37
38         # если пролетели по всем точкам
39         if current_mission_point >= len(mission_point):
40             break
41
42     pioneer.land()
```



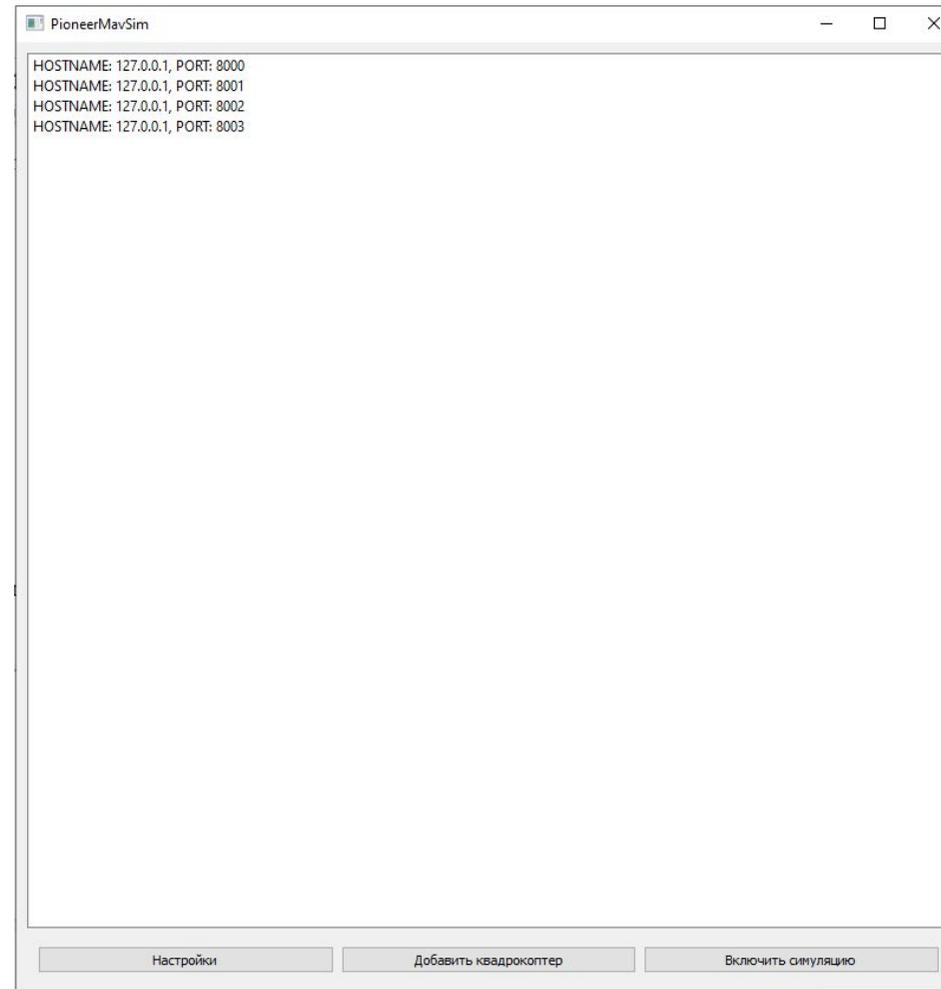
Скачать симулятор можно двумя путями:

1) <https://github.com/IlyaDanilenko/PioneerMavSim>

2) [https://drive.google.com/drive/folders/1ZWmGmYnSYFo23\\_4dlzImnPZmT1ZQPkjr?usp=sharing](https://drive.google.com/drive/folders/1ZWmGmYnSYFo23_4dlzImnPZmT1ZQPkjr?usp=sharing)

**ВАЖНО:**

Для работы симулятора необходим **python 3.9**



# Домашнее задание

## *обязательно*



Установить PyCharm.

Установить библиотеку для работы с Пионером с гита.

Написать скрипт полета выполняющий следующие действие: Взлет -> Полет в точку (3, 3, 2) с посадкой -> Полет в точку (7,7,1.5) с посадкой -> Полет в точку (1,1, 1.5) с посадкой.

\*Для тестов создать объект класса Pioneer(method=2, pioneer\_ip="127.0.0.1", pioneer\_mavlink\_port=8000)



**Самарин Алексей**  
**Программист отдела образовательных**  
**проектов ГК "Геоскан"**

**Telegram - канал**