

---

GEOSCAN

# ГЕОСКАН ПИОНЕР

РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ





---

## Содержание:

---

<b>1 Сборка «Пионера»</b>	<b>3</b>
1.1 Сборка рамы	5
1.2 Сборка защиты	8
1.3 Первое включение	11
<b>2 Настройки</b>	<b>13</b>
2.1 Обновление прошивки	13
2.2 Настройка пульта	14
2.3 Связь пульта с приемником	15
2.4 Настройка параметров автопилота	15
2.5 Работа с логами автопилота	17
<b>3 Полёт</b>	<b>21</b>
3.1 Симулятор	21
3.2 Аккумуляторная батарея	23
3.3 Безопасность при полете	24
3.4 Пульт радиоуправления	25
3.5 Управление	27
3.6 Подготовка к полету - первый старт.	28
<b>4 Дополнительные модули</b>	<b>31</b>
4.1 Плата подключения дополнительных модулей	32
4.2 Модуль захвата груза	34
4.3 Модуль навигации GPS Глонасс	36
4.4 Программируемая камера OpenMV	39
4.5 Модуль оптического позиционирования	43
4.6 Модуль LED	46
4.7 Подвес для камеры GoPro	49
4.8 Камера для фото и видео съемки	50
4.9 FPV комплект	51
4.10 Бортовой модуль навигации в помещении	51
<b>5 Программирование</b>	<b>55</b>
5.1 Визуальное программирование TRIK	55
5.2 Pioneer Station	58
5.3 Программирование на Lua	63
<b>6 Система навигации в помещении Геоскан Локус</b>	<b>73</b>
6.1 Установка полетной зоны	74
6.2 Обновление прошивки системы навигации	76

<b>7</b>	<b>FAQ</b>	<b>79</b>
<b>8</b>	<b>Changelog</b>	<b>81</b>
8.1	20/06/2019 . . . . .	81
8.2	11/02/2019 . . . . .	81
8.3	1/02/2019 . . . . .	82



---

[Вернуться на сайт «Пионер»](#)

Геоскан «Пионер» - это многофункциональный квадрокоптер для новичков. Его можно использовать как учебную платформу в школе, авиамodelьной секции, кружке робототехники или дома.



С «Пионером» вы сможете:

- изучить основы робототехники и авионики;
- узнать, как устроен квадрокоптер;
- научиться управлять им;
- адаптировать платформу под разные задачи;
- освоить азы программирования.

---

**Важно:** Возможности квадрокоптера можно расширить, подключая дополнительные модули.

---

Таблица 1: Технические характеристики

Параметр	Значение
Тип	квадрокоптер
Продолжительность полета	до 17 минут
Скорость полета	до 65 км/ч
Масса квадрокоптера	230 г
Размеры	290 x 290 x 120 мм
Двигатели	бесколлекторные 1306 3100KV
Аккумуляторная батарея	LiPo 2S 1300мАч 9,62 Втч
Максимальная дальность полета	500 м
Допустимая скорость ветра	до 5 м/с
Температура эксплуатации	От 0 до +40 °С

Прочитайте это руководство, чтобы собрать и запустить «Пионер» в воздух. Раздел «Программирование» также поможет вам решать с с более сложные задачи.



## Сборка «Пионера»

**Важно:** В наборе «Пионер» есть все элементы и инструменты для сборки и запуска квадрокоптера.

### Воздушные винты

х4



Воздушный винт  
правого вращения

х4



Воздушный винт  
левого вращения

### Базовая плата

х1



### Защита

х8



Дуга защиты

х8



Перемычка защиты

х4

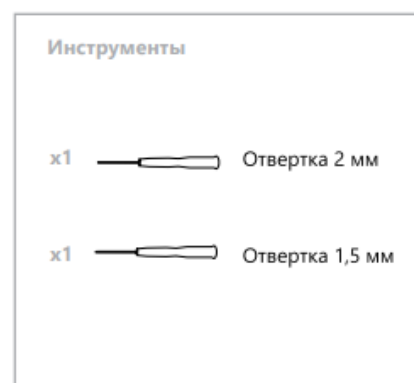
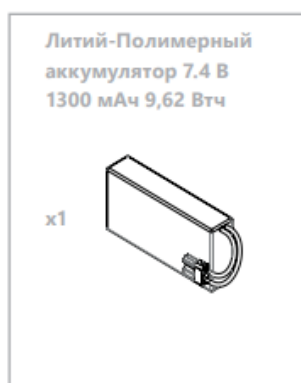
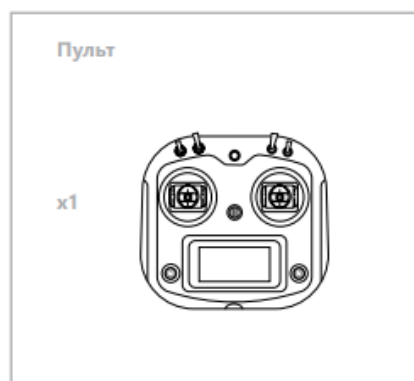
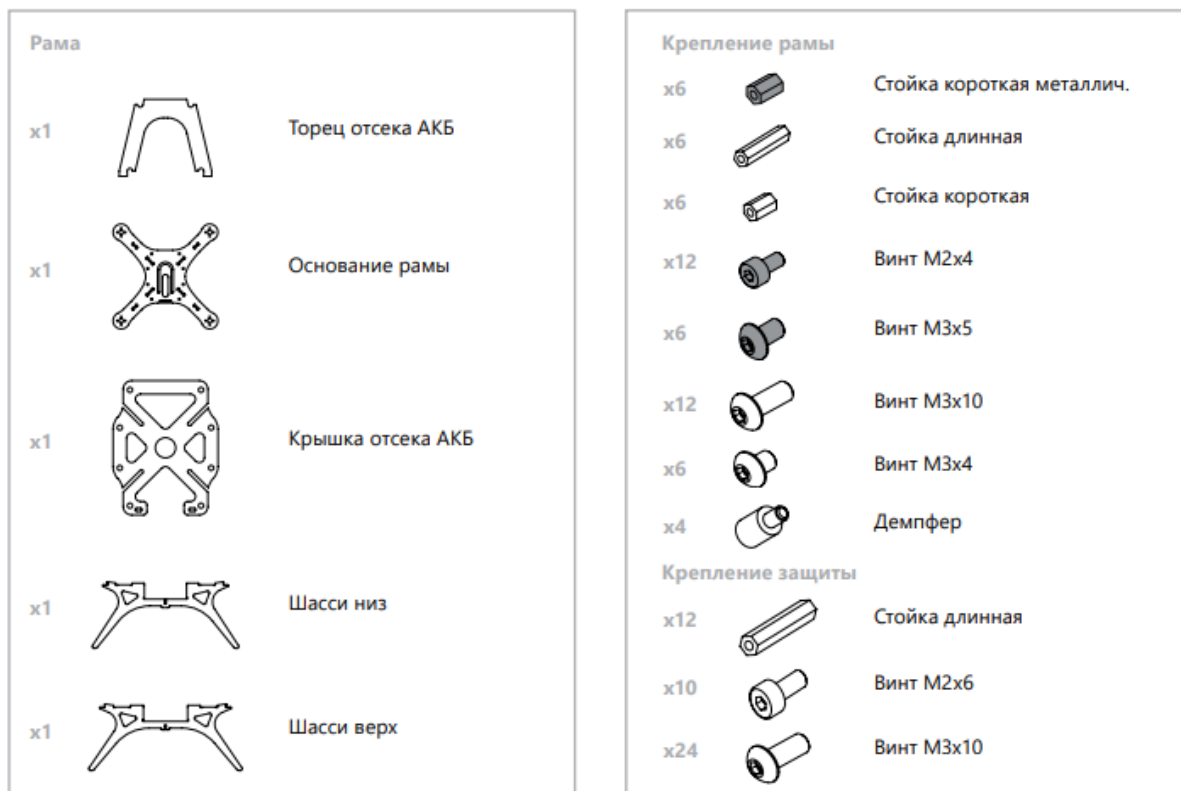


Основание защиты

### USB кабель

х1





Собирайте «Пионер» в два этапа:

*Сборка рамы*

*Сборка защиты*

В разделе *Дополнительные модули* подробно рассказано о том, как их подключить и запрограммировать.

## 1.1 Сборка рамы

В наборе «Пионер» есть все элементы и инструменты для сборки.

### Шаг 1

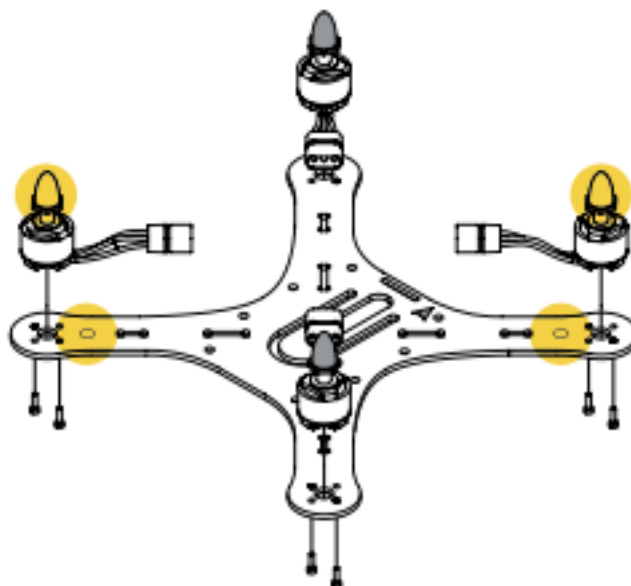


Рис. 1: Используются: основание рамы, мотор правого вращения - 2 шт., мотор левого вращения - 2 шт., винт M2x4 - 8 шт.

Места установки моторов с серебристыми гайками отмечены на основании рамы белыми кругами. Следуя цветовой маркировке, установите моторы на основание рамы и закрепите каждый мотор двумя винтами, расположив их по диагонали.

### Шаг 2

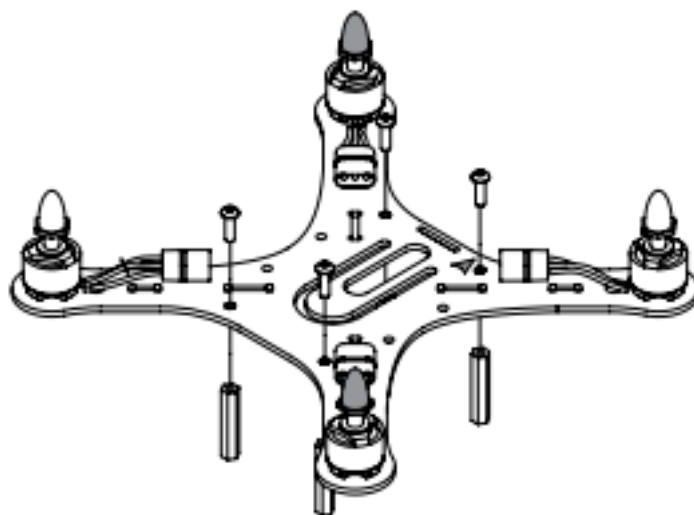


Рис. 2: Используются: узел, собранный на шаге 1, стойки длинные 4 - шт., винты M3x10 - 4 шт.

Установите стойки с противоположной от моторов стороны основания рамы и закрепите их винтами M3x10, как показано на рисунке.

### Шаг 3

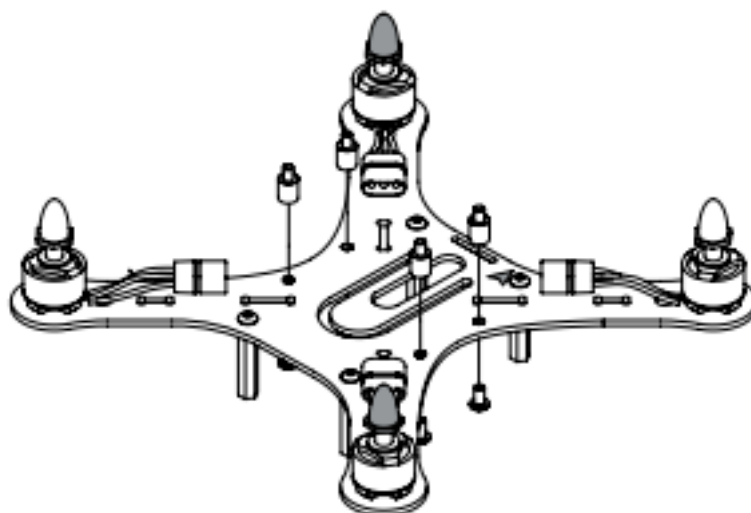


Рис. 3: Используются: узел, собранный на шаге 2, демпферы – 4 шт., винты М3х4 - 4 шт.

Установите демпферы на основание рамы со стороны моторов и закрепите их винтами М3х4, как показано на рисунке.

#### Шаг 4

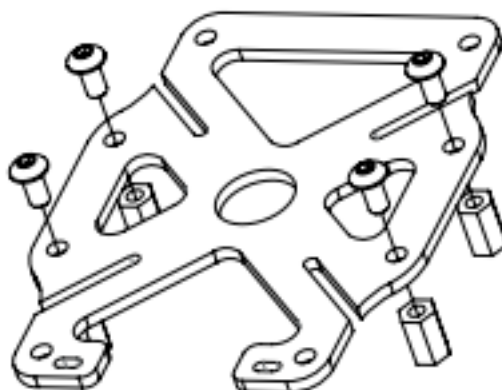


Рис. 4: Используются: крышка отсека АКБ, стойки 8 мм - 4 шт., винты М3х5 - 4 шт.

С помощью винтов М3х5 закрепите стойки на крышке отсека АКБ, как показано на рисунке. Эти стойки могут потребоваться в дальнейшем для крепления дополнительных модулей.

#### Шаг 5

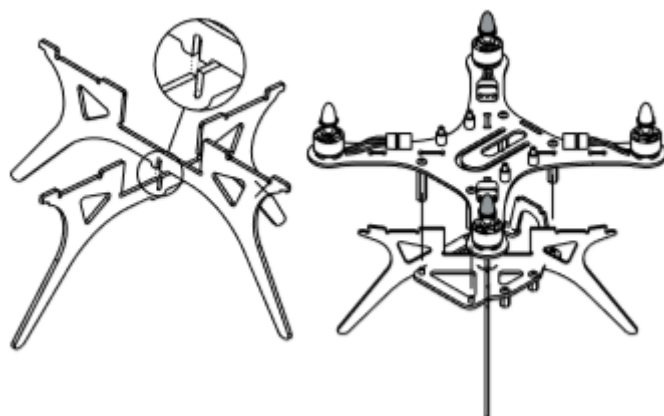


Рис. 5: Используются: узел, собранный на шаге 3, шасси верх, шасси низ, торец отсека АКБ.

Вставьте шасси и торец отсека АКБ в соответствующие пазы, расположенные на основании рамы.

### Шаг 6

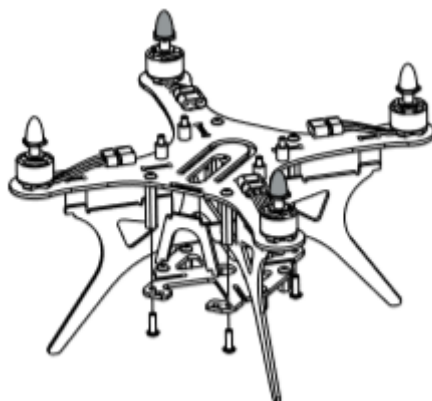


Рис. 6: Используются: узел, собранный на шаге 4, узел, собранный на шаге 5, винты М3х10.

С помощью винтов М3х10 соедините узел, собранный на шаге 4, с узлом, собранным на шаге 5, как показано на рисунке.

### Шаг 7

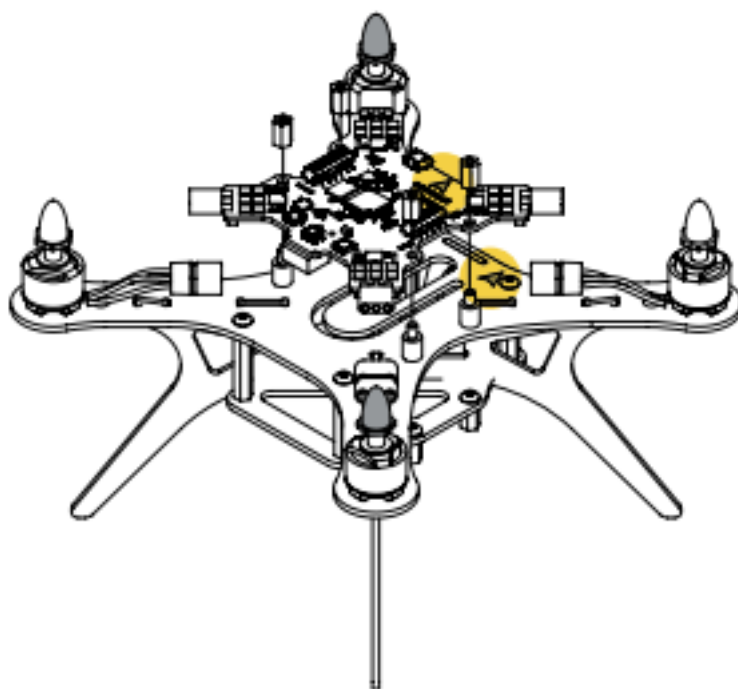


Рис. 7: Используются: узел, собранный на шаге 6, стойки короткие металлические 4 шт., базовая плата.

Установите плату на демпферы так, чтобы направления стрелок на плате и на основании рамы совпадали. Зафиксируйте плату с помощью стоек. Подключите разъемы моторов к соответствующим разъемам на плате. Разъемы имеют ключ для соблюдения полярности, будьте внимательны и не прикладывайте излишних усилий при соединении разъемов.

Собрав раму «Пионера», перейдите к [Сборка защиты](#)

## 1.2 Сборка защиты

### Шаг 8



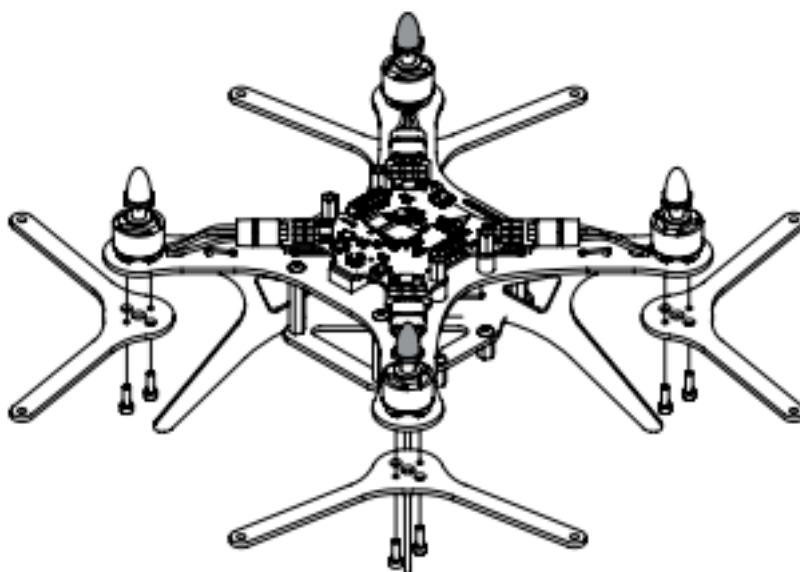


Рис. 8: Используются: узел, собранный на шаге 7, основания защиты - 4 шт., винты М2х6 - 8 шт.

Установите основания защиты и закрепите их винтами М2х6, как показано на рисунке.

#### Шаг 9

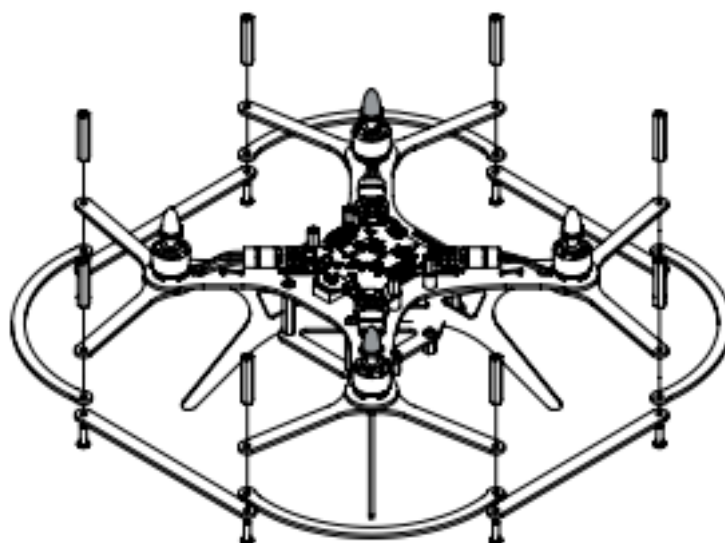


Рис. 9: Используются: узел, собранный на шаге 8, дуга защиты - 8 шт., перемычка защиты - 8 шт., винты М3х10 - 8 шт., стойки длинные- 8 шт.

С помощью винтов М3х10 закрепите на основаниях защиты дуги защиты, перемычки защиты и стойки, как показано на рисунке.

#### Шаг 10

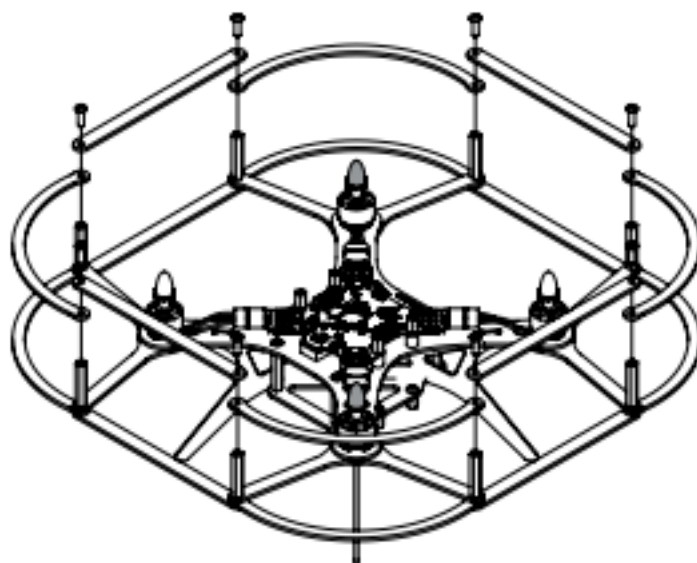


Рис. 10: Используются: узел, собранный на шаге 9, дуга защиты - 8 шт., перемычка защиты - 8 шт., винты М3х10 - 8 шт.

С помощью винтов М3х10 закрепите на стойках защиты дуги защиты и перемычки защиты, как показано на рисунке.

#### Шаг 11

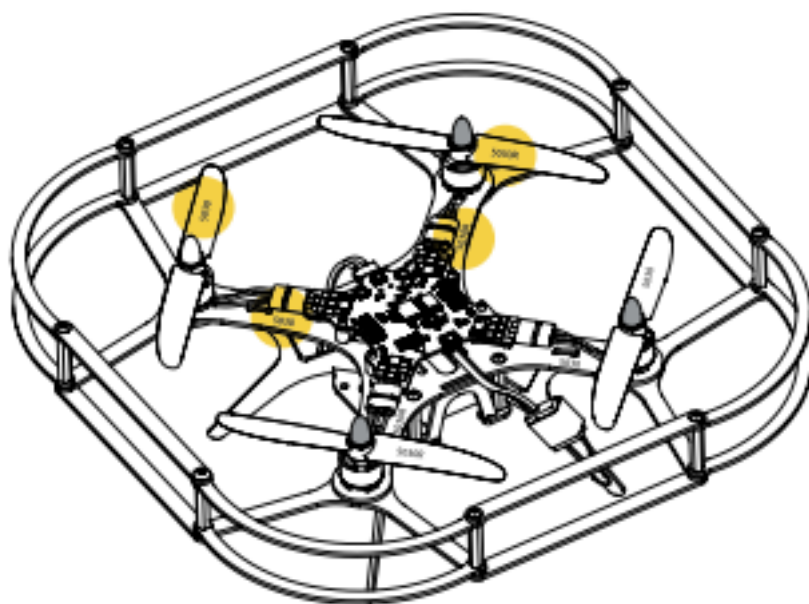


Рис. 11: Используются: узел, собранный на шаге 10, винт воздушный 5030 - 2 шт., винт воздушный 5030R - 2 шт.

Обратите внимание: черные гайки имеют левую резьбу! Надписи на винтах (5030 или 5030R) должны совпадать с надписями на раме. Открутите гайки с осей моторов, установите воздушные винты (надписи на винтах должны быть сверху) и закрепите их гайками. Серебристые гайки закручиваются по часовой стрелке (правая резьба), черные гайки – против часовой стрелки (левая резьба).

#### Шаг 12

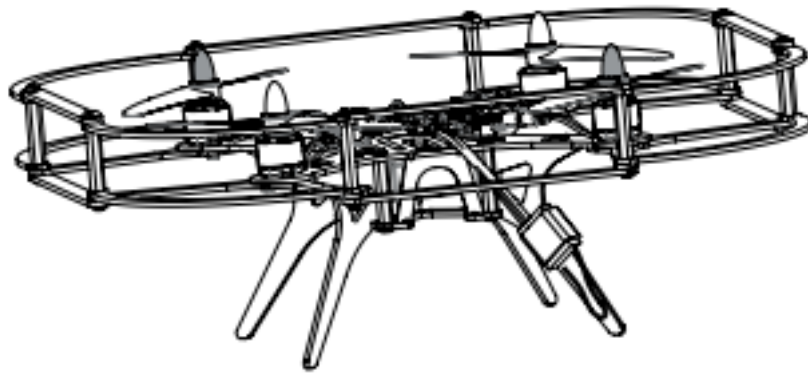


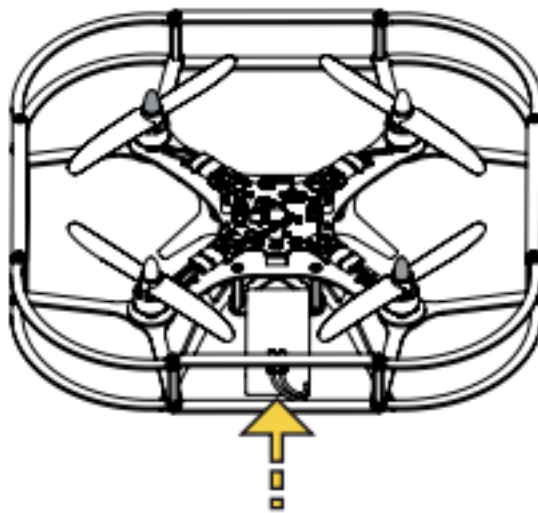
Рис. 12: Используются: узел, собранный на шаге 12, приемник (находится в упаковке пульта управления).

Закрепите приемник на ножке шасси с помощью эластичного фиксатора. Вставьте разъем кабеля в разъем PPM на базовой плате (Разъем PPM находится под разъемом micro-USB).

Квадрокоптер собран, чтобы запустить его прочитайте раздел *Первое включение*

### 1.3 Первое включение

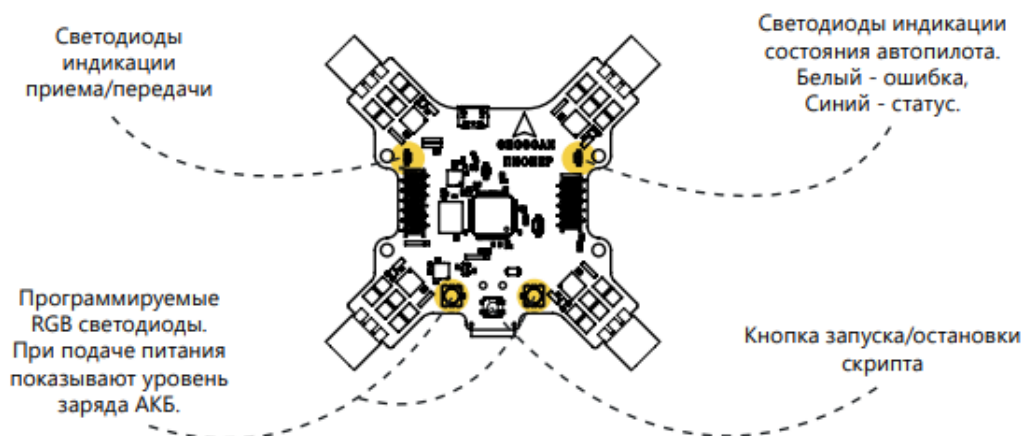
Вставьте аккумулятор в раму «Пионера» и включите силовой(желтый) разъем в желтое гнездо на базовой плате.



---

#### Важно:

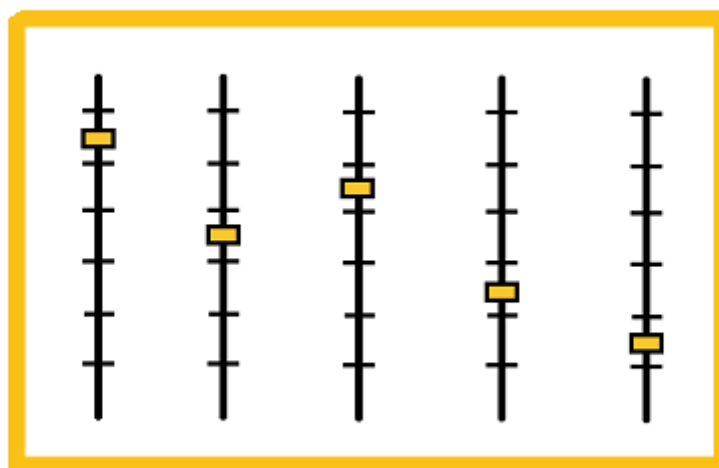
- При подключении аккумулятора квадрокоптер издает звуковой сигнал и мигает светодиодами, отображая уровень заряда аккумулятора. Красный цвет – аккумулятор разряжен, зеленый – заряжен
- Медленно мигает синий светодиод – готов к эксплуатации
- Быстро мигает синий светодиод – на квадрокоптере запущено выполнение скрипта
- Быстро мигает белый светодиод - ошибка
- При падении напряжения аккумулятора ниже допустимого квадрокоптер издает звуковые сигналы



**Внимание:**

**«Пионер» оснащен системой отключения моторов при нештатных ситуациях:**

- при наклоне квадрокоптера более чем на 90°;
- при ударе или падении;



Прошивку «Пионера» при необходимости можно обновить. В этом разделе содержится подробная инструкция, которая вам поможет.

В случае замены приемника управления или пульта, перейдите в раздел [Связь пульта с приемником](#)

Также вы можете заново [настроить пульт](#), если что-то сбилось.

Инструкция по точной настройке «Пионера» находится в разделе [Настройка параметров автопилота](#)

В разделе [Работа с логами автопилота](#) описано как просматривать данные с «черного ящика» квадрокоптера после завершения полёта.

## 2.1 Обновление прошивки

Для корректной работы всех систем «Пионера» рекомендуется устанавливать последнюю актуальную версию прошивки. Для этого на вашем компьютере должна быть установлена программа

---

Pioneer Station. Если у вас ее нет, нажмите [скачать установщик Pioneer Station](#), загрузка начнется автоматически. После этого необходимо запустить скачанный exe файл.

---

**Примечание:** Если программа не запускается, попробуйте [скачать 32bit версию](#), распаковать архив и запустить run.bat.

---

Далее нужно перевести «Пионер» в режим обновления. Подключите квадрокоптер к компьютеру через usb кабель. Запустите Pioneer Station, нажмите на значок квадрокоптера в левом верхнем углу окна программы. В выпадающем меню нажмите «обновление прошивки» и следуйте инструкциям помощника.

---

**Примечание:** Если при выборе устройства квадрокоптер не отображается в окне, попробуйте переподключить его и перезапустить программу. Если это не помогло, попробуйте [скачать и установить драйвер порта](#), после чего переподключите «Пионер» и перезапустите программу.

---

При выборе источника прошивки рекомендуется выбрать пункт «встроенный», так вы установите последнюю актуальную версию.

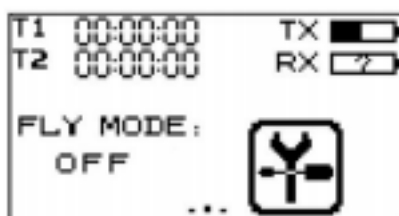
Дождитесь окончания прошивки, это может занять несколько минут. После завершения нажмите «ОК», квадрокоптер перейдет в обычный режим работы.

Если Вы столкнулись с проблемой обновления прошивки или работы «Пионера», напишите в техподдержку Геоскан [support@geoscan.aero](mailto:support@geoscan.aero)

## 2.2 Настройка пульта

Для настройки пульта выполните следующие действия шаг за шагом:

1. Вставьте батарейки и включите пульт дистанционного управления. Для этого нажмите одновременно две кнопки включения питания и удерживайте их до включения экрана пульта.
2. При включении на экране должна появиться картинка:



Если на экране отображается надпись «Warning!!! Place all switches in their up position», приведите все переключатели в верхнее положение, а стик газ/курс в нижнее положение.

3. Войдите в меню настроек, нажав на сенсорном экране значок настроек:



4. В меню настроек найдите вкладку FUNCTION. В ней производится настройка пульта управления. Вкладка SYSTEM отвечает за настройку самой модели.

### Настройки вкладки FUNCTION для работы с «Пионером»:

1. REVERSE → Ch2 и Ch4 –Rev
2. AUX. CHANNELS → Channel 5 → → В открывшемся окне CH TYPE выбрать тип переключателя SWx → нажать на надпись SwA, и в открывшемся меню выбрать SwC.

3. AUX. CHANNELS → Channel 6 → → В открывшемся окне CH TYPE выбрать тип переключателя SWx → нажать на надпись SwA, и в открывшемся меню выбрать SwD.
4. AUX. CHANNELS → Channel 7 → → В открывшемся окне CH TYPE выбрать тип переключателя SWx → нажать на надпись SwA, и в открывшемся меню выбрать SwB.

**Во вкладке SYSTEM:**

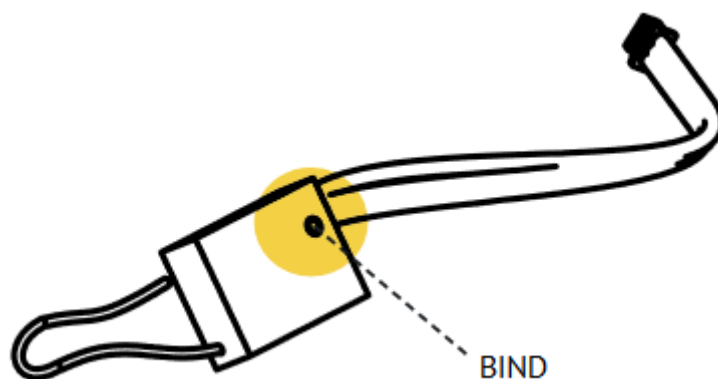
5. OUTPUT MODE → Output → PPM
6. STICKS MODE → M2 (Mode 2)

Для дальнейшей настройки перейдите в раздел *Связь пульта с приемником*

## 2.3 Связь пульта с приемником

Для того, чтобы пульт и приемник могли работать в паре, необходимо выполнить процедуру привязки. Инструкция к ней - ниже.

**Внимание:** Чтобы избежать ситуации одновременного управления несколькими квадрокоптерами с одного пульта, разрешается связывать только один пульт с одним приемником.



1. Включите пульт дистанционного управления
2. Войдите в меню настроек
3. Во вкладке SYSTEM выберите пункт Rx Bind. Появится надпись «Binding to RX...»
4. Найдите на приемнике кнопку с обозначением «BIND», нажмите и удерживайте ее.
5. Держа кнопку зажатой, подайте питание на квадрокоптер, подключив аккумулятор
6. Выйдите из настройки Rx Bind. Квадрокоптер должен отреагировать на это действие звуковым сигналом.

Если моторы не вращаются, и квадрокоптер никак не реагирует на сигналы с пульта, повторите процедуру привязки или обратитесь в техническую поддержку по адресу: [support@geoscan.aero](mailto:support@geoscan.aero)

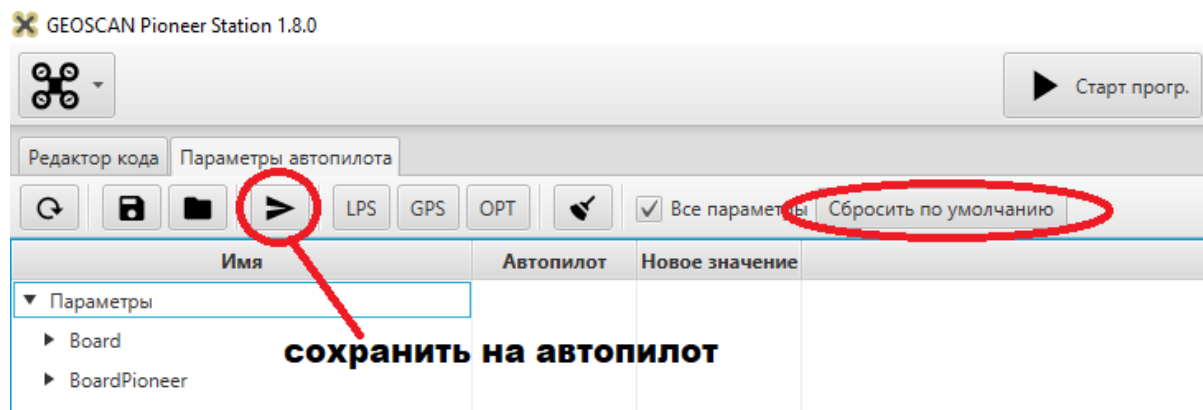
**Важно:** Процедура привязки работает только для приемника FlySky-A8S. При включении пульта квадрокоптер издаст звуковой сигнал.

## 2.4 Настройка параметров автопилота

Параметры полета квадрокоптера в автоматическом режиме можно тонко настроить, используя программу *Pioneer Station*. Для этого подключите «Пионер» к компьютеру следуя [инструкции](#) и

перейдите из вкладки «редактор кода» в «настройки параметров автопилота»

Для настройки «Пионер» должен быть подключен к компьютеру, его текущие параметры должны отображаться в правой части окна.



Перейдите во вкладку «параметры автопилота». В первом столбце таблицы приведены названия параметров. Узнать, за что каждый из них отвечает, можно, посмотрев в крайний правый столбец. Во втором столбце отображаются текущие значения, присвоенные каждому из параметров. Если вы хотите поменять какое-либо из них, введите новое значение в столбец справа от текущего значения и нажмите Enter. Завершив редактирование, не забудьте сохранить изменения, нажав кнопку **сохранить изменения на автопилот**.

В верхней части окна также присутствуют кнопки **LPS**, **GPS** и **OPT**. Каждая из них загружает в квадрокоптер стандартный набор параметров для полета, соответственно, в локальной системе позиционирования, с использованием модуля GPS/ГЛОНАСС и модуля оптического позиционирования.

**Примечание:** Если вы хотите вернуться к заводским настройкам, нажмите кнопку **сбросить по умолчанию** над таблицей и подождите пока «Пионер» перезапустится.

### 2.4.1 Настройка параметров при замене двигателей

Если вы заменили двигатели на Пионере, они могут отличаться по электрическим характеристикам и не пройдут предстартовые проверки при подготовке к взлёту. За неё отвечают параметры:

- **Copter\_motorCheckTime** - время проверки скорости оборотов двигателей на старте в секундах. Для отключения проверки установите значение параметра 0
- **Copter\_startRpmMax** - максимальное количество об/мин для проверки на старте
- **Copter\_startRpmMin** - минимальное количество об/мин для проверки на старте
- **Copter\_startRpmSigma** - максимальное расхождение между измеренными оборотами двигателей, при превышении которого считается, что двигатели работают неравномерно.

При получении команды на взлет (первые 5 секунд) проводится ещё одна проверка:

- **Copter\_stallRpm** - предельная величина оборотов, после которой считается, что произошел срыв синхронизации, и двигатели отключаются. Для нестандартных двигателей можно использовать максимально возможные холостые обороты двигателя (для этого нужно умножить kv двигателя на напряжение аккумулятора)

В случае замены двигателей на более мощные вам также может понадобиться настроить PID коэффициенты. Неправильные PID - причина заметных вибраций и колебаний копитера в полёте.

Чтобы настроить PID коэффициенты, используйте следующие параметры:



- **Copter\_xyRate\_ki** - интегральная часть регулятора. Его нужно уменьшить, если квадрокоптер нехотя отзывается на управление, и увеличить, если заметны низкочастотные колебания.
- **Copter\_xyRate\_kp** - пропорциональная часть регулятора. Его можно уменьшить, если присутствует заметное перерегулирование и высокочастотные колебания (коптер дрожит во время зависания в одной точке). Если квадрокоптер плохо отзывается на управление, значение параметра нужно увеличить.

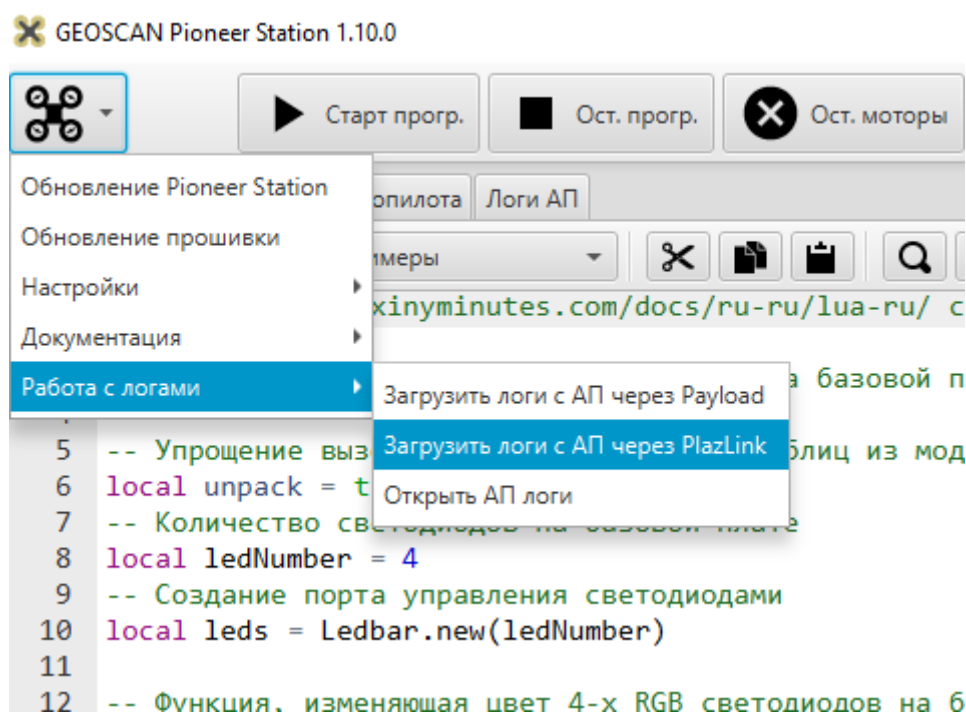
**Примечание:** Для оценки допустимых оборотов, запишите логи полёта в ручном режиме управления и просмотрите график **rpm** в канале данных **motor-x**. Подробнее в разделе [Работа с логами автопилота](#)

## 2.5 Работа с логами автопилота

С момента запуска моторов автопилот Пионера записывает параметры полёта по отдельным каналам и сохраняет их в файл, который можно скачать и проанализировать. Запись лога прекращается в момент отключения моторов. При повторном запуске моторов предыдущий лог стирается, и начинается запись нового.

### 2.5.1 Скачивание файла лога

Чтобы скачать лог, подключите Пионер к компьютеру и запустите Pioneer Station. Установите соединение с коптером по USB, после чего в меню откройте вкладку **Работа с логами - Загрузить логи с АП через PlazLink**.



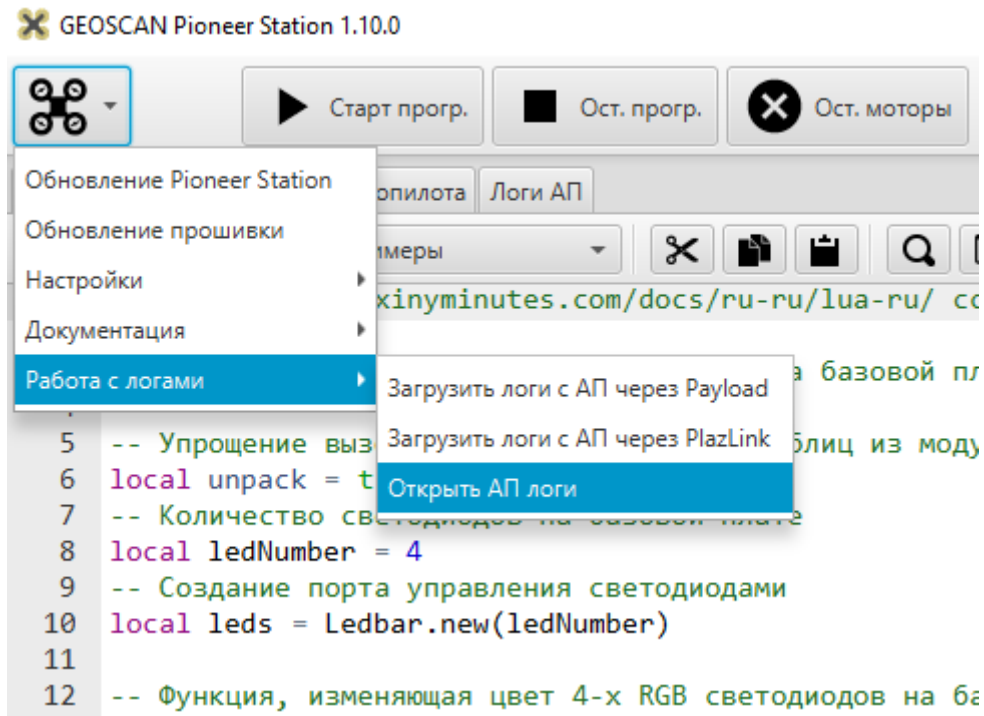
В открывшемся окне выберите папку и название, под которым файл будет сохранён. После этого начнётся скачивание файла с Пионера. После завершения квадрокоптер перезагрузится.

**Примечание:** Старые версии прошивки не поддерживают скачивание лога через протокол PlazLink. В таком случае выберите пункт меню **Работа с логами - Загрузить логи с АП через Payload**. Следуя указаниям помощника, нажмите и удерживайте кнопку **Старт** на плате Пионера

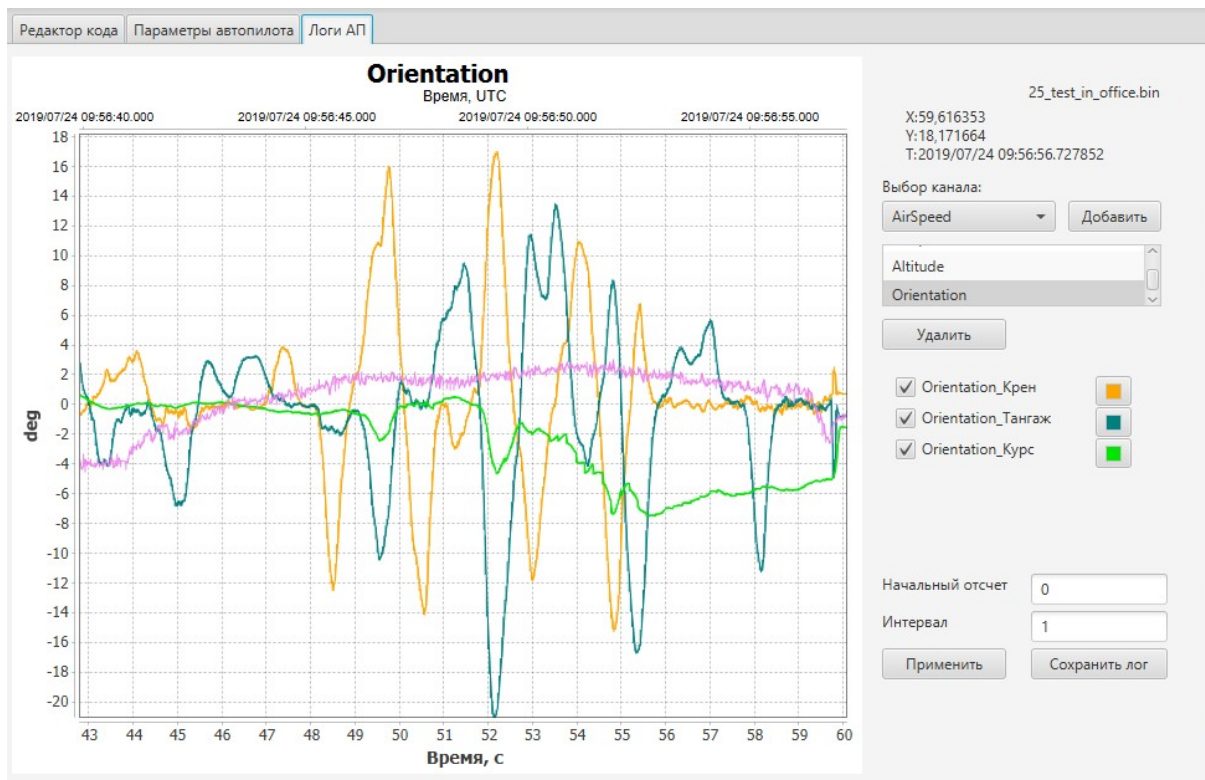
и подключите его к компьютеру. После этого нажмите кнопку **Далее**, выберите папку назначения и запустите скачивание лога.

## 2.5.2 Просмотр файла лога

Чтобы открыть файла лога, откройте вкладку меню **Работа с логами - Открыть АП логи**.



В открывшейся вкладке **Логи АП** вы можете просматривать графики каждого из каналов.



- Выберите нужный канал и добавьте его в окно просмотра.

- Настраивайте отображение информации при помощи чекбоксов и цветовой палитры графиков.
- Используйте правую кнопку мыши и колесико прокрутки для навигации и масштабирования.
- Чтобы регулировать масштаб только по оси времени, зажмите и удерживайте **Ctrl** и используйте колесико прокрутки.
- Чтобы очистить окно просмотра, выберите ненужные каналы и нажмите **Удалить**. Это действие не удаляет лог, а только отменяет отображение графика.

Канал	Описание
Altitude	высота по барометру в метрах
ControlData	управляющие сигналы от автопилота
IntPowerStatus	данные об аккумуляторе
IntStaticPressure	данные с барометра
LpsState	данные ультразвуковой системы навигации
MagHeading	данные магнитометра
Motor-6X	данные регуляторов двигателей
NavPosition	данные о положении по GPS
NavPrecision	данные о точности по GPS
NavSatInfo	данные о количестве спутников и статусе
NavVelocity	данные о скоростях по 3-м осям по GPS
OpticalFlow	данные модуля оптического потока
Orientation	ориентация коптера в пространстве
RawAccelGyroData	RAW данные акселерометра, гироскопа и их температура
RemoteControlData	данные управления, поступившие на Пионер с пульта



Этот раздел поможет вам освоить управление квадрокоптером и совершить первый успешный полет. Прочитайте его прежде чем перейти к самостоятельным полетам.

### 3.1 Симулятор

Если у вас нет опыта управления квадрокоптером, а также если вы хотите совершенствовать свои навыки, начните с полетов в симуляторе. Так вы освоите управление без риска для квадрокоптера, себя и окружающих.

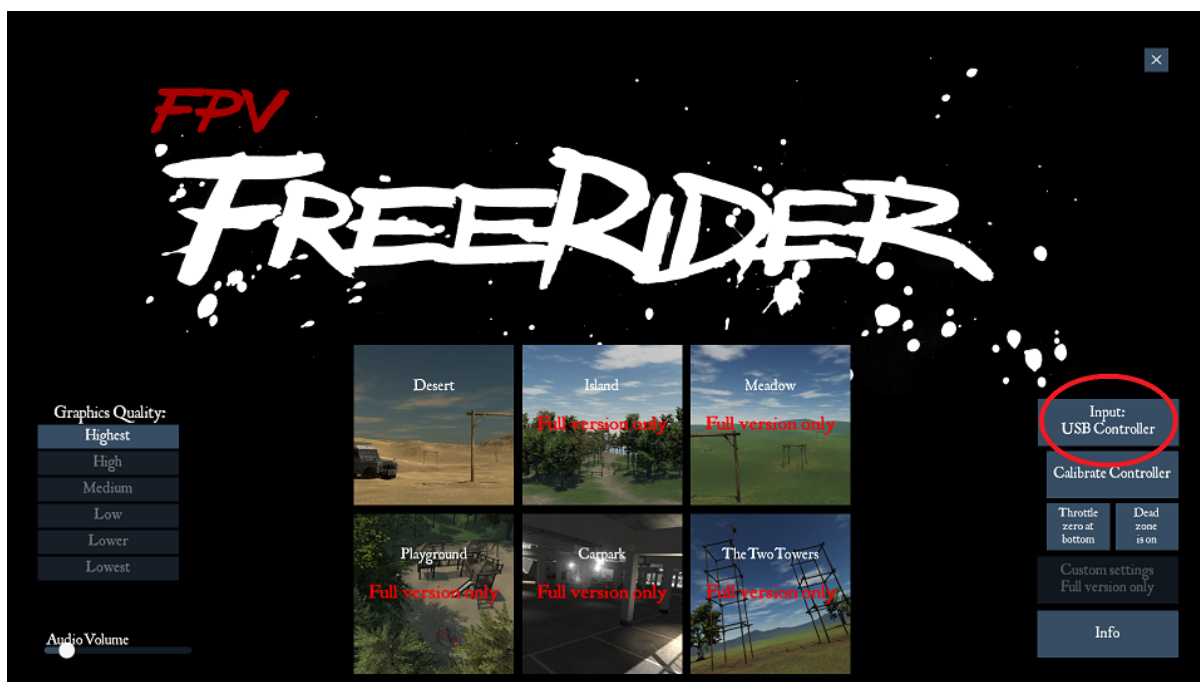
Подключите пульт FlySky F6S к компьютеру проводом microUSB, и вы сможете тренироваться с ним в любом симуляторе.

В разделе *Пульт радиоуправления* смотрите за что отвечают кнопки и стики.

Из доступных симуляторов полета на квадрокоптере большинство - платные. Для получения начальных навыков пилотирования достаточно бесплатных аналогов или демо-версий.

Мы предлагаем попробовать FPVFreerider. Скачать бесплатную версию можно [здесь](#), прокрутите страницу до Download demo и выберите версию для Вашей ОС.

Установив программу, подключите пульт к компьютеру, включите его и запустите симулятор. Все тумблеры на пульте должны быть в крайнем верхнем положении.



Выберите управление через USB controller и перейдите в пункт **calibrate controller**. Далее следуйте указаниям на экране, поочередно приводя соответствующий стик в крайнее положение и нажимая ОК. По завершении калибровки проверьте правильность реакции программы на движения стиков, подправьте ошибки с помощью ползунков **trim**. Виртуальный пульт должен в точности повторять ваши движения, от этого зависит удобство управления.

Закончив калибровку, откройте доступную карту для полета.

Вначале обычно рекомендуют научиться управлять квадрокоптером с видом от третьего лица (Line of sight) и в режиме стабилизации. Для этого щелчком мыши по кнопкам в левом верхнем углу экрана выберите режим self-level, low rates и camera, как на скриншоте:



**Перейдем к полетам:**

- Плавно толкните левый стик вперед. Когда квадрокоптер оторвется от поверхности, постарайтесь удерживать его на небольшой высоте, а затем приземлиться, плавно возвращая стик



---

в нижнее положение. Научитесь чувствовать момент взлета и касания.

- удерживая квадрокоптер на одной высоте, толкните и отпустите правый стик в любом направлении. Обратите внимание, что при наклоне квадрокоптер не только разгоняется, но и теряет высоту. Чтобы компенсировать снижение, добавьте оборотов левым стиком. Научитесь летать в разных направлениях без потери высоты и не упуская квадрокоптер из виду. Если хотите начать заново, нажмите кнопку Reset в левой части экрана.
- Наклон левого стика в стороны управляет поворотом квадрокоптера. При изменении направления управлять полетом становится сложнее, т.к. надо учитывать в какую сторону «смотрит» аппарат. Стандартное упражнение на такой случай - полет по квадрату. Попробуйте летать по воображаемому квадрату небольшой площади, разворачивая коптер на 90 градусов в каждом угле. Пролетев против часовой стрелки, смените направление. Постепенно вы научитесь контролировать направление полета.

Управление квадрокоптером - это навык, требующий практики. Осваивая полет в симуляторе, вы сможете чувствовать свой прогресс. Научившись лететь в выбранном направлении, не допуская падений и не теряя квадрокоптер из виду, можете смело переходить к практике на «Пионере».

---

**Примечание:** Также в симуляторе можно научиться управлять коптером в режиме вида от первого лица (FPV) с курсовой камеры. Для этого щелчком мыши переключите кнопку Camera в режим FPV.

---

## 3.2 Аккумуляторная батарея



В качестве источника энергии «Пионер» использует LiPo аккумулятор емкостью 1300 мАч. Батареи такого типа легкие и мощные, но могут быть пожароопасны при неправильной эксплуатации. Прочитав этот раздел, вы научитесь работать с LiPo аккумуляторами.

### Зарядка

1. Для зарядки выберете негорючую поверхность вдали от легковоспламеняемых предметов (шторы, ковры, мягкая мебель)
2. Осмотрите аккумулятор, на нем не должно быть вмятин, бугров, разрывов оболочки.
3. Подключите зарядное устройство к сети 220 В.
4. Подключите балансировочный (маленький) вывод аккумулятора к разъему зарядного устройства.

- Индикаторы заряда должны гореть красным цветом. Зеленый цвет индикатора указывает на то, что ячейка заряжена.
- Дождитесь полной зарядки аккумулятора и отключите его от зарядного устройства. Нельзя оставлять заряжающийся аккумулятор без присмотра!

**Примечание:** Если вы планируете хранить аккумулятор, не используя, дольше месяца - разрядите его примерно наполовину и дозаряжайте до того же состояния раз в 3-4 месяца. При таком режиме аккумулятор не выйдет из строя при переразряде.

**Внимание:** Не заряжайте и не используйте поврежденный аккумулятор. До момента утилизации его следует хранить отдельно от исправных, вдали от горючих предметов и поверхностей. Используйте металлический контейнер (например, кастрюлю с крышкой) или специальный огнестойкий чехол.

### Эксплуатация

- При установке аккумулятора в раму следите, чтобы пропеллеры не задевали провода.
- Подключите аккумулятор к квадрокоптеру перед взлетом и отключите сразу после посадки.
- Не допускайте короткого замыкания между контактами аккумулятора.
- Перед взлетом и после посадки осмотрите аккумулятор. На нём не должно быть повреждений, вмятин, проколов и разрывов оболочки. Недопустим нагрев батареи выше 50°C и разбухание оболочки.

**Опасно:** Если в полете аккумулятор был поврежден, **не пытайтесь немедленно достать его из рамы!** От изгиба и дополнительного притока воздуха батарея может вспыхнуть. При появлении дыма или пламени из квадрокоптера немедленно совершите посадку на негорючую поверхность.

**Недопустимо тушить аккумулятор водой!** Для тушения можно использовать плотную ткань, песок, землю, углекислотный или порошковый огнетушитель. Если тушить нечем - дождитесь пока батарея догорит сама и утилизируйте то что осталось.

Продукты горения LiPo аккумулятора **крайне токсичны**, не вдыхайте образовавшийся дым.

## 3.3 Безопасность при полете

«Пионер» предназначен для эксплуатации в помещении и на открытом воздухе при температуре окружающей среды от 0 до +40 °C и относительной влажности не более 80%. Квадрокоптер не предназначен для полетов в условиях выпадения дождя, снега и при порывистом ветре. Для безопасной эксплуатации «Пионера» соблюдайте следующие правила:

- Перед полетом убедитесь, что над квадрокоптером нет препятствий (лампы, ветки)
- Обращайте внимание на провода и тросы, натянутые на высоте. По возможности не взлетайте в таких местах.
- Сильный или порывистый ветер может опрокинуть или сбить квадрокоптер с курса. Допустимой считается скорость ветра до 5 м/с. Помните, что при наборе высоты ветер может менять скорость и направление.
- Не запускайте выполнение программы с элементами полета при подключенном проводе USB. Если необходима калибровка ESC - демонтируйте пропеллеры.
- Перед каждым полетом проверяйте целостность пропеллеров. Полет с гнутым или разрушенным пропеллером может привести к повреждению квадрокоптера.



---

**Примечание:** Повреждение винта в полете можно распознать по изменению звука и характера полета. Заметив эти признаки, посадите квадрокоптер и осмотрите винты, при необходимости - замените.

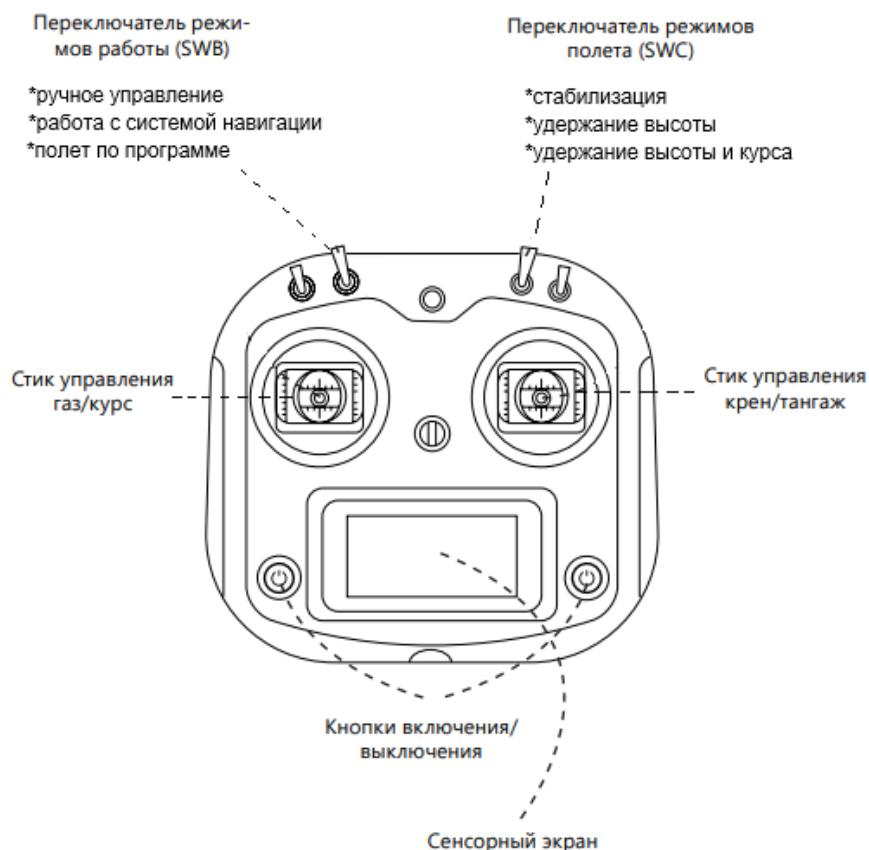
---

### 3.4 Пульт радиоуправления

Для управления «Пионером» используйте аппаратуру FlySky i6S.



- По умолчанию пульт готов к эксплуатации и не требует дополнительной настройки или привязки.
- Квадрокоптер совместим с пультами управления, которые поддерживают протокол PPM и работают с приемником FlySky.
- Используйте стики для управления полетом квадрокоптера.
- Используйте переключатель режимов для выбора режима полета квадрокоптера.
- Чтобы включить пульт, нажмите и удерживайте кнопки включения/выключения одновременно.



### Изменение режима работы пульта:

- Режим управления с пульта – Квадрокоптер под вашим контролем. Выберите режим полета переключателем SWC.
- В режиме работы с системой навигации или с GPS – квадрокоптер управляется с пульта, но сохранит высоту и положение если отпустить стики.
- В режиме полета по программе Пионер выполняет загруженную миссию. Вы всегда можете взять управление на себя и переключиться в режим ручного управления.

### Изменение режима полета:

- Режим стабилизации (stabilize)– Стик газа отвечает за изменение уровня мощности, подаваемой на моторы.
- Режим удержания высоты (althold) – квадрокоптер автоматически удерживает высоту, стик газа отвечает за изменение вертикальной скорости.
- Режим удержания высоты и курса (headless) - управление высотой происходит как в предыдущем режиме. Отличие в том, что квадрокоптер запоминает направление курса в момент включения моторов. Если вращать квадрокоптер по углу рыскания, направления вперед/назад и влево/вправо будут неизменны.

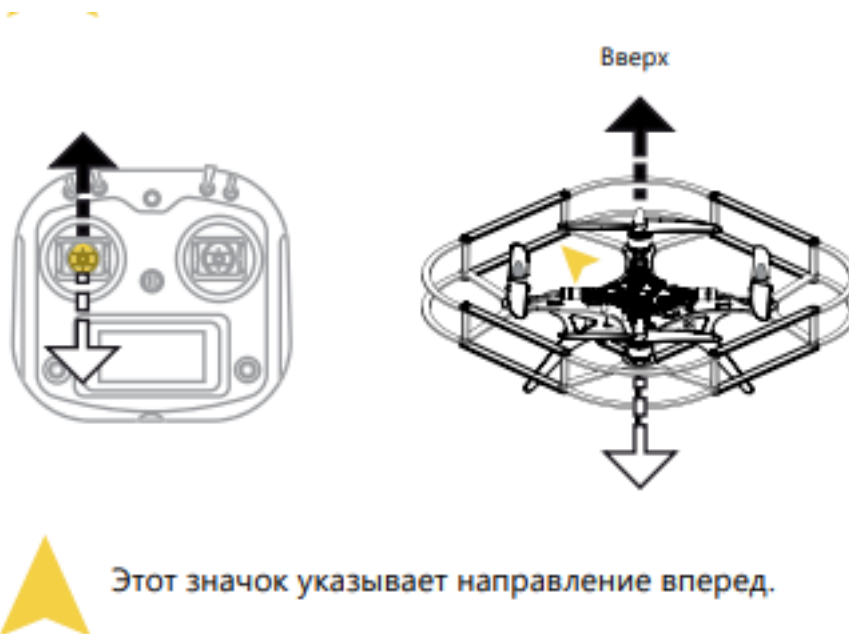
**Внимание:** Для определения высоты полетный контроллер использует барометр, который измеряет атмосферное давление. Если давление воздуха меняется, аппарат будет реагировать на изменение давления, а не на изменения фактической высоты.

---

## 3.5 Управление

Для управления «Пионером» с пульта управления следуйте правилам приведенным ниже:

### Управление высотой

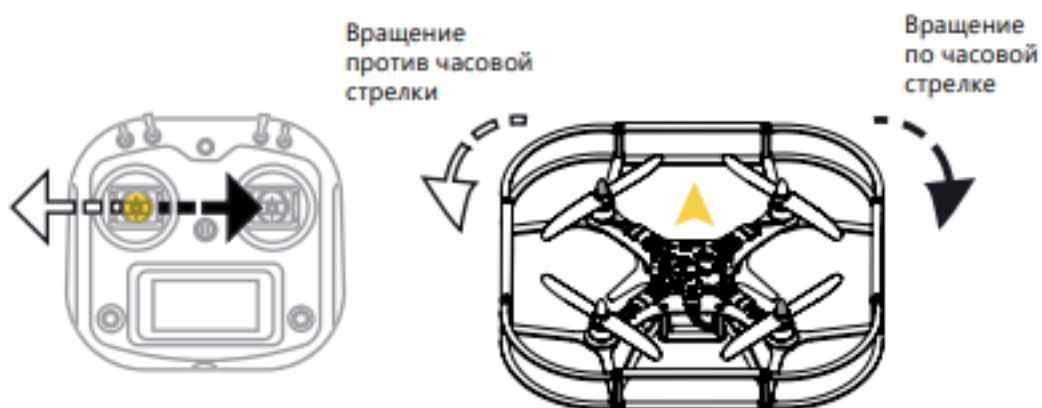


---

**Важно:** При перемещении стика газ/курс от себя квадрокоптер набирает высоту, при перемещении на себя – снижается.

---

### Вращение вокруг своей оси

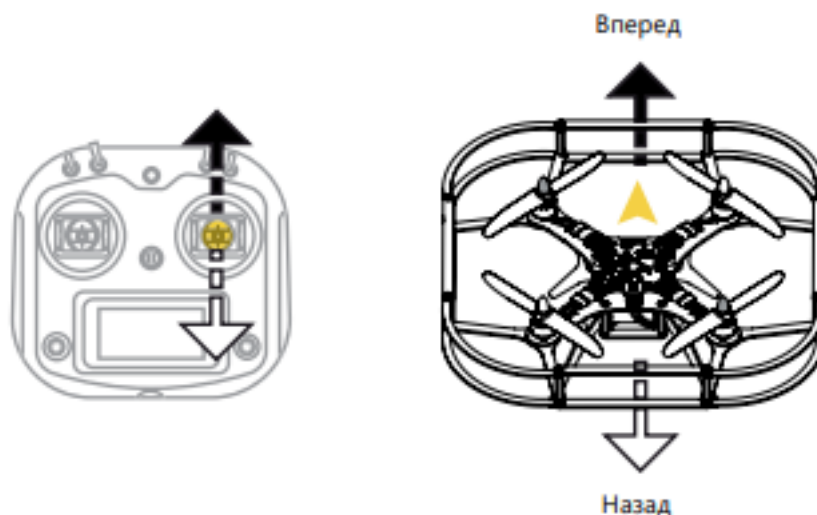


---

**Важно:** При перемещении стика газ/курс влево/вправо квадрокоптер поворачивается вокруг своей оси.

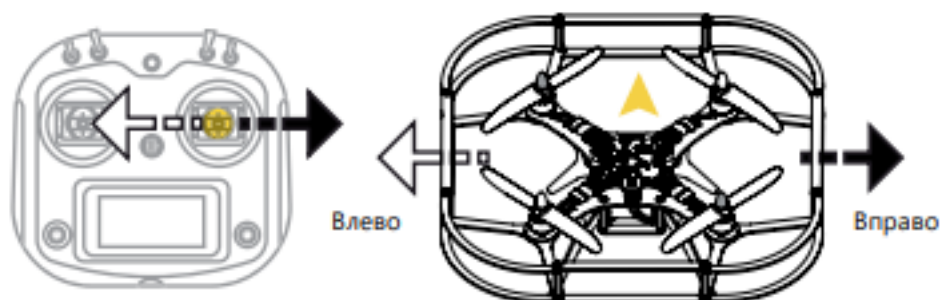
---

### Полет вперед/назад



**Важно:** При перемещении стика крен/тангаж от себя, квадрокоптер наклоняется вперед и летит вперед. При перемещении стика крен/тангаж на себя, квадрокоптер наклоняется назад и летит назад.

#### Полет боком влево/вправо



**Важно:** При перемещении стика крен/тангаж влево/вправо, квадрокоптер летит боком влево/вправо.

**Внимание:** Управляйте квадрокоптером плавно, не дергайте стики пульта слишком резко или сразу в крайнее положение. Пульт управления держите двумя руками, пальцы обеих рук всегда держатся за стики.

Для выполнения первого полета перейдите в раздел *Подготовка к полету - первый старт*.

### 3.6 Подготовка к полету - первый старт.

Перед полетом прочтите раздел *Управление*

**Для успешного старта:**

1. Полностью зарядите аккумулятор квадрокоптера и проверьте заряд батареек в пульте.

2. Убедитесь что на квадрокоптере нет заметных повреждений.
3. Установите стик газ/курс в положение минимум - до упора на себя.
4. Включите пульт управления;
5. До упора вставьте аккумулятор в отсек в нижней части квадрокоптера. Убедитесь, что он не выпадает при наклоне квадрокоптера.
6. Подключите аккумулятор к базовой плате.
7. Прозвучит звуковой сигнал и на квадрокоптере загорятся светодиоды.
8. Дождитесь пока светодиод статуса не начнет мигать синим.
9. Запустите моторы. Для этого до упора зажмите левый стик в правый нижний угол на 2-3 секунды.
10. Дождитесь пока все винты начнут вращаться и плавно взлетайте. Во время полета соблюдайте правила эксплуатации. Не допускайте падения квадрокоптера.
11. После полета остановите моторы. Для этого до упора зажмите левый стик в левый нижний угол на 2-3 секунды.

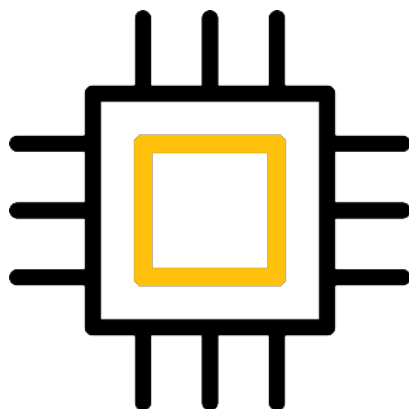




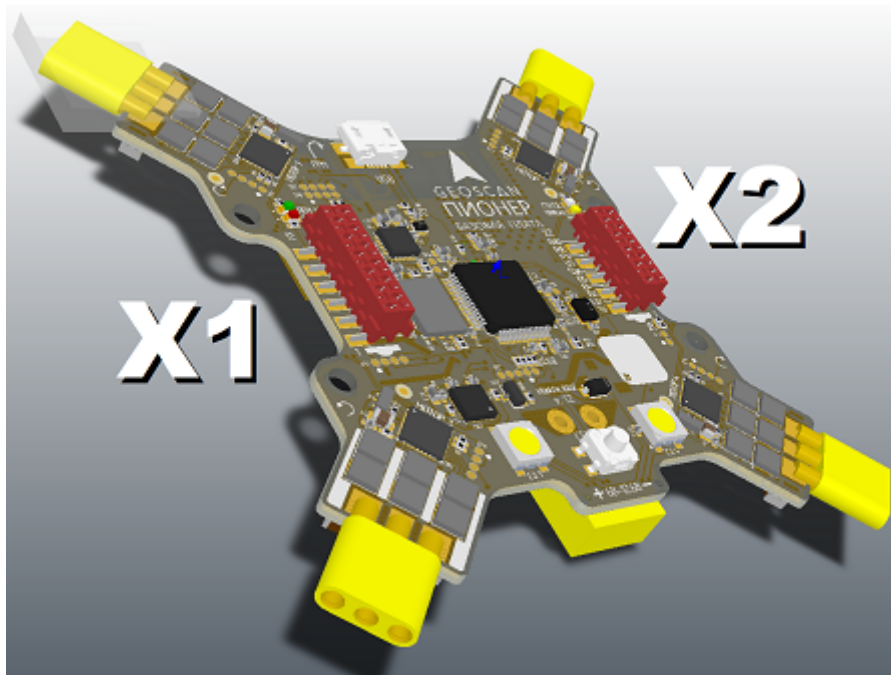
---

Дополнительные модули

---



Подключите к «Пионеру» дополнительные модули, чтобы получить доступ к новым функциям.



Для подключения используются разъёмы X1 и X2 на верхней части платы. Большинство модулей монтируются на плате подключения дополнительных модулей и соединяются с базовой платой шлейфами. Модули GPS и навигации в помещении крепятся напрямую на базовую плату «Пионера».

---

**Примечание:** Разъёмы отличаются по количеству контактов и габаритам. Не подключайте шлейф в несоответствующий разъём. Соблюдайте правильность подключения, используя маркировочный выступ на «вилке» шлейфа и не прикладывайте излишнюю силу.

---

В следующих подразделах подробно рассказано, как подключить и запрограммировать каждый модуль.

---

**Примечание:** Настройка работы модулей выполняется с помощью [Pioneer Station](#) перед полетом.

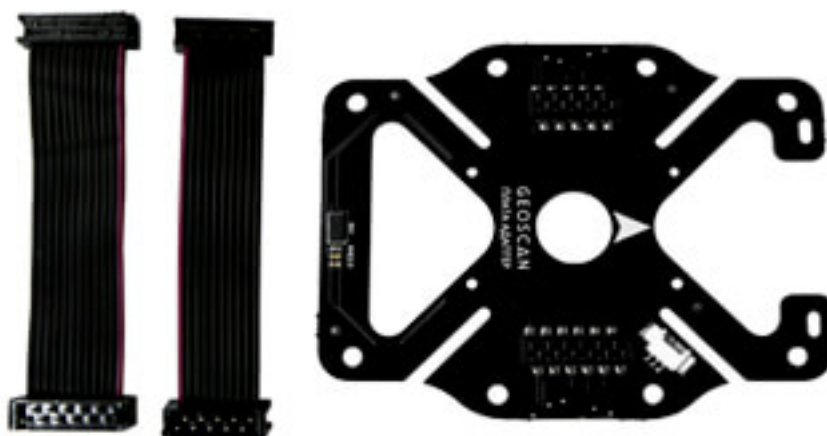
---

**Помните, что подключенные модули потребляют дополнительную энергию и снижают время работы аккумулятора. Рассчитывайте полетное время, исходя из режима работы и числа установленных модулей.**

## 4.1 Плата подключения дополнительных модулей

Используйте эту плату, чтобы подключать больше модулей к «Пионеру».





При установке ориентируйте плату так, чтобы стрелка на ней указывала в ту же сторону, что и стрелка на основной плате квадрокоптера. Плата крепится на нижнюю часть квадрокоптера вместо крышки отсека аккумулятора через стойки винтами М3.

Сверху на плате расположены два разъема. Используйте их, чтобы подключать шлейфы, идущие от базовой платы. На нижней поверхности - два разъема для подключения модулей. Модули подключаются «насквозь» через плату.

Также плата подключения дополнительных модулей оснащена лазерным дальномером. Он позволяет точно определять высоту над уровнем пола в пределах 0 - 1,5 м. Дальномер активируется в режиме удержания высоты. При этом «Пионер» будет сохранять одинаковое расстояние от поверхности, даже если она неровная. Пролетая над препятствием или впадиной, квадрокоптер будет менять высоту. Если закрыть датчик рукой или другим предметом, «Пионер» будет набирать высоту до тех пор, пока расстояние до препятствия не вернется к прежнему значению, или пока квадрокоптер не упрется в потолок.

Данные с дальномера можно считывать и использовать для управления квадрокоптером. В качестве примера приведена программа, меняющая цвет светодиодов на «Пионере» в зависимости от расстояния до земли. Используйте Pioneer Station, чтобы [загрузить программу](#) на «Пионер».

**Примечание:** Дальномер неактивен в режиме полета по GPS. Если программа не работает, откройте вкладку «[параметры автопилота](#)» в Pioneer Station и выберите набор параметров «LPS» или «OPT». Также можно вручную поменять значение параметра BoardPioneer\_modules\_gnss на 0,0.

```
-- https://learnxinyminutes.com/docs/ru-ru/lua-ru/
--
local range = Sensors.range
--
local ledNumber = 4
--
local leds = Ledbar.new(ledNumber)
```

(continues on next page)

```
--
local function changeColor(red, green, blue)
    -- 4-
    for i = 0, ledNumber - 1, 1 do
        leds:set(i, red, green, blue)
    end
end

function callback(event)
end

--
timerRangeRead = Timer.new(0.1, function ()
    -- ( 1- )
    distance = range()
    r, g, b = 0, 0, 0
    -- 8.19
    if (distance >= 8.19) then
        --
        r = 1
    else
        -- (~1.5 - )
        g = math.abs(distance / 1.5)
    end
    --
    changeColor(r, g, b)
end)
--
timerRangeRead:start()
```

## 4.2 Модуль захвата груза



Используйте модуль захвата груза для работы с предметами, обладающими магнитным свойства-

ми. Модуль оснащен электромагнитом на гибком подвесе и RGB светодиодами.

Модуль монтируется на плату подключения дополнительных модулей с помощью 4х винтов М3, при этом обеспечивается надежное соединение разъемов.

### 4.2.1 Пример

Модулем удобно управлять по 8 каналу с помощью двухпозиционного переключателя на пульте. Для настройки канала на нем необходимо во вкладке function найти пункт Aux. channels, выбрать channel 8 и установить переключатель SwA (крайний левый) как управляющий. Теперь нужно загрузить на «Пионер» программу для модуля груза. Ниже - пример Lua программы. После её загрузки на «Пионер» вы сможете управлять работой магнита с помощью крайнего левого тумблера на пульте.

```
--          -   PC3          1.2
local magnet = Gpio.new(Gpio.C, 3, Gpio.OUTPUT)
--          -   PA1          1.1 (
↪ )
-- local magnet = Gpio.new(Gpio.A, 1, Gpio.OUTPUT)

--          (4          4          )
local led_number = 8
--
local leds = Ledbar.new(led_number)
--          (          )
local magnet_state = true

--          ,
local function setLed(state)
  if (state == true) then
    color = {1,1,1} --          ,
  else
    color = {0,0,0} --          ,          (          )
  end
  for i = 4, led_number - 1, 1 do --          4
    leds:set(i, table.unpack(color))
  end
end

--
local function toggleMagnet()
  if (magnet_state == true) then --          ,
    magnet:reset()
  else --          ,
    magnet:set()
  end
  magnet_state = not magnet_state --
end

--
function callback(event)
end

--          ,
cargoTimer = Timer.new(1, function ()
  toggleMagnet()
  setLed(magnet_state)
end)
--
cargoTimer:start()
```

## 4.3 Модуль навигации GPS Глонасс



Модуль позволяет квадрокоптеру отслеживать свое текущее положение и скорость. Для большей точности позиционирования антенне модуля необходим обзор неба, не используйте режим GPS в помещении.

Модуль GPS/Глонасс устанавливается непосредственно на главную плату квадрокоптера сверху и крепится к ней винтами М3. При подключении дополнительных модулей на нижнюю плату расширения их шлейфы стыкуются с разъемами на верхней части модуля GPS, таким образом подключаясь к основной плате «сквозь» модуль GPS.

Модуль также оснащен компасом для точной ориентации в пространстве. Показания компаса могут искажаться вблизи массивных металлических объектов и зданий.

При подключении к Pioneer Station убедитесь, что в «параметрах автопилота» выбран режим позиционирования по GPS, в противном случае активируйте его нажатием на кнопку. В стандартном режиме подключения работу компаса можно наблюдать на панели авиагоризонта.

---

**Примечание:** При первом подключении рекомендуется убедиться в правильной работе компаса. Сориентируйте Пионер по оси «север-юг» и убедитесь, что показания в Pioneer Station соответствуют сторонам света при повороте квадрокоптера на 90 градусов. Если погрешность превышает 10 градусов, выберите «калибровка - компаса» в меню Pioneer Station и следуйте указаниям помощника.

---

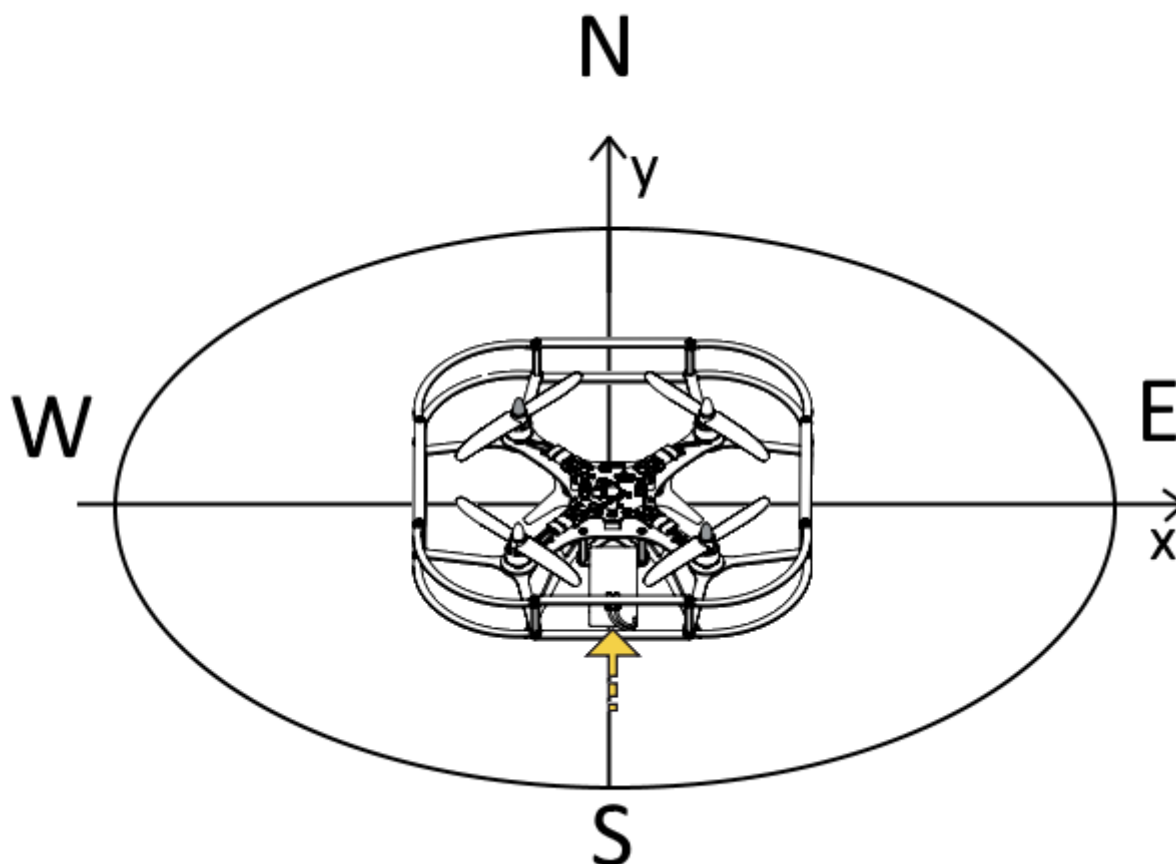
Также в реальном времени отображается количество спутников, доступных для работы. Чем больше спутников «видит» модуль, тем выше точность позиционирования. При холодном старте (первом включении модуля в новой точке) синхронизация занимает от одной до трех минут. При успешной синхронизации зеленый светодиод на модуле горит постоянно. Теперь, если выключить питание «Пионера» и через некоторое время снова включить в той же точке, синхронизация произойдет значительно быстрее.

LED индикация	состояние	что делать
горят красный и зеленый	неверные параметры	переключить параметры на GPS
горит красный	поиск спутников	ждать 1-3 минуты, перезагрузить
горит зеленый	спутники найдены	можно лететь

Управляя квадрокоптером с пульта, выберите режим работы с системой навигации (тумблер SwB в среднем положении), и «Пионер» будет использовать подключенный модуль GPS, что сделает полет более стабильным.

---

При программировании полёта используется команда `ap.goToLocalPoint(x, y, z)`. При этом ось  $x$  направлена на восток, ось  $y$  - на север. Значение  $z$  определяет высоту относительно точки старта. Все значения указываются в метрах.



Также можно использовать команду `ap.goToPoint(x, y, z)`. Здесь  $x$ ,  $y$  это географические координаты (широта, долгота) точки, в которую полетит Пионер.  $Z$  - высота в этой точке относительно точки старта.

---

**Примечание:** Если указанная точка старта будет дальше, чем в 500 метрах от фактического местоположения, квадрокоптер откажется взлетать. Также, параметры `Flight_com_flyAreaSize` и `Flight_com_maxAltitude` ограничивают расстояние и высоту, на которую коптер может улететь от точки старта. Подробнее - в описании параметров в Pioneer Station.

---

#### 4.3.1 Пример

Ниже приведен пример программы, использующий модуль GPS. Загрузите программу на Пионер, найдите подходящую площадку и подключите аккумулятор. Когда светодиод на модуле перестанет мигать, переключите SwB в нижнее положение и запустите выполнение программы, нажав кнопку «Старт» на базовой плате квадрокоптера. Через 5 секунд Пионер взлетит, пролетит на 10 метров на восток и вернется на точку старта.

Пионер определяет свое местоположение в момент запуска моторов. Эта точка принимается за начало координат при выполнении команды `ap.goToLocalPoint`.

Неточность позиционирования с GPS и ошибка определения высоты могут достигать 3 метров. Учитывайте это при составлении своих полетных программ.

```

--
local unpack = table.unpack
--
local ledNumber = 4
--
local leds = Ledbar.new(ledNumber)
--
local function changeColor(color)
  --
  -- 4-
  for i=0, ledNumber - 1, 1 do
    leds:set(i, unpack(color))
  end
end

--
-- RGB
-- changeColor
local colors = {
  {1, 0, 0}, --
  {0, 1, 0}, --
  {0, 0, 1}, --
  {1, 1, 0}, --
  {1, 0, 1}, --
  {0, 1, 1}, --
  {1, 1, 1}, --
  {0, 0, 0} -- /
}

--
-- {x,y,z}
local points = {
  {0, 0, 0.7},
  {0, 1, 0.7},
  {0.5, 1, 0.7},
  {0.5, 0, 0.7}
}

--
local curr_point = 1

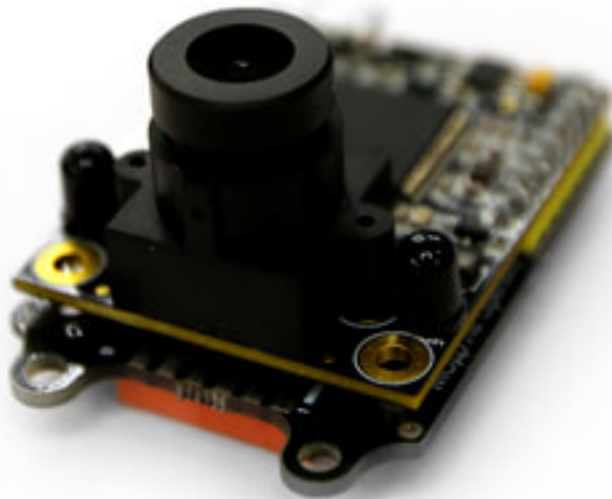
--
local function nextPoint()
  --
  -- . % - , # - . ,
  --
  curr_color = ((curr_point - 1) % (#colors - 2)) + 1
  --
  changeColor(colors[curr_color])
  --
  if(curr_point <= #points) then
    Timer.callLater(1, function()
      --
      ap.goToLocalPoint(unpack(points[curr_point]))
      --
      curr_point = curr_point + 1
    end)
  --
  else
    Timer.callLater(1, function()
      --
      ap.push(Ev.MCE_LANDING)
    end)
  end
end
end

```

(continues on next page)

```
--  
function callback(event)  
    --                               Flight_com_takeoffAlt,  
    if(event == Ev.TAKEOFF_COMPLETE) then  
        nextPoint()  
    end  
    --  
    if(event == Ev.POINT_REACHED) then  
        nextPoint()  
    end  
    --  
    if (event == Ev.COPTER_LANDED) then  
        changeColor(colors[8])  
    end  
end  
  
--  
ap.push(Ev.MCE_PREFLIGHT)  
--  
changeColor(colors[7])  
-- , 2  
Timer.callLater(2, function() ap.push(Ev.MCE_TAKEOFF) end)
```

## 4.4 Программируемая камера OpenMV



OpenMV - это программируемая камера. Она может самостоятельно обрабатывать видео поток и посылать команды управления на «Пионер».

Для установки камеры на квадрокоптер сначала установите модуль OpenMV. Закрепите его на 4 винта М3 и подключите оба разъема. Сама камера крепится к модулю на два 8-пиновых разъема. Для дополнительной надежности можно соединить платы камеры и модуля через две стойки с

винтами.

Подробнее о настройке и программировании камеры - на [странице OpenMV](#)

#### 4.4.1 Пример

Модуль OpenMV взаимодействует с базовой платой «Пионера» через интерфейс UART. Для его настройки нужно написать две отдельных программы для квадрокоптера и камеры. Для настройки соединения на «Пионере» используйте следующий код:

```
--          Uart
local uartNum = 4 --          Uart          (USART4)
local baudRate = 9600 --
local stopBits = 1
local parity = Uart.PARITY_NONE
local uart = Uart.new(uartNum, baudRate, parity, stopBits)
```

Теперь нужно настроить протокол обмена на самой камере. Для этого [скачайте и запустите OpenMV IDE](#) для своей ОС. Подключите камеру к компьютеру кабелем micro-USB и нажмите кнопку «Соединить» в левом нижнем углу OpenMV IDE. Теперь написанный в текстовом поле код можно загружать непосредственно в камеру, нажав кнопку «запустить» в левом нижнем углу.

Код для инициализации интерфейса UART на камере:

```
uart = UART(3)
uart.init(9600, bits=8, parity=None, stop=1, timeout_char=1000)
```

Протокол обмена между камерой и «Пионером» создан.

Ниже - пример программы, которая меняет яркость свечения светодиодов на «Пионере» в зависимости от яркости картинки, принимаемой камерой OpenMV. В коде обеих программ (для «Пионера» и OpenMV) уже содержатся блоки создания UART интерфейса.

Код для «Пионера»

```
-- https://learnxinyminutes.com/docs/ru-ru/lua-ru/          LUA
--
local ledNumber = 4
--
local leds = Ledbar.new(ledNumber)

--          ,          4- RGB
local function changeColor(red, green, blue)
  for i=0, ledNumber - 1, 1 do
    leds:set(i, red, green, blue)
  end
end

--          ,
local function emergency()
  takePhotoTimer:stop()
  --
  Timer.callLater(1, function () changeColor(1, 0, 0) end)
end

--
function callback(event)
  --
  if (event == Ev.LOW_VOLTAGE2) then
    emergency()
  end
end
```

(continues on next page)



```

changeColor(1, 0, 0) -- red

--      Uart
local uartNum = 4 --      Uart      (USART4)
local baudRate = 9600 --
local dataBits = 8
local stopBits = 1
local parity = Uart.PARITY_NONE
local uart = Uart.new(uartNum, baudRate, parity, stopBits) --

changeColor(1, 0, 1) --purple

local N = 10
local i = 7
local strUnpack = string.unpack
function getData() --
  i = i + 1
  if (i == N + 1) then i = 0 end
  buf = uart:read(uart:bytesToRead()) or '0'
  if (#buf == 0) then buf = '\0' end
  leds:set(1, 0, i/N, 0.5 - 0.5*i/10)
  if (strUnpack ~= nil) then
    local b = strUnpack("B", buf)
    return b --
  else
    return 20
  end
end

local takerFunction = function () --
  local intensity = getData() / 100.0
  changeColor(intensity, intensity, intensity)
end
local interval = 0.1
getMeasureTimer = Timer.new(interval, takerFunction) --
getMeasureTimer:start()

changeColor(1, 0.2, 0) -- orange

```

#### Код для OpenMV

```

# Hello World Example
#
# Welcome to the OpenMV IDE! Click on the green run arrow button below to run the script!

from pyb import UART, LED

import sensor, lcd, image, time, utime

ledBlue = LED(2)
ledGreen = LED(3)

ledBlue.on()
sensor.reset() # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.LCD) # Set frame size to QVGA (320x240)
sensor.skip_frames(100) # Wait for settings take effect.
clock = time.clock() # Create a clock object to track the FPS.

```

(continues on next page)

```

lcd.init()
#lcd.set_backlight(True)
ledBlue.off()

#Init uart

uart = UART(3)
uart.init(9600, bits=8, parity=None, stop=1, timeout_char=1000) # init with given parameters

M_LED_COUNT = 10
led_counter = M_LED_COUNT
led_mode = 0
while(True):
    clock.tick()                # Update the FPS clock.
    clk = utime.ticks_ms()
    img = sensor.snapshot()     # Take a picture and return the image.
    #print(clock.fps())         # Note: OpenMV Cam runs about half as fast when connected
                                # to the IDE. The FPS should increase once disconnected.

    for r in img.find_rects(threshold = 40000):
        img.draw_rectangle(r.x(), r.y(), r.w(), r.h(), (255, 0, 0))
        for p in r.corners():
            img.draw_circle(p[0], p[1], 5, color = (0, 255, 0))
        print(r)

    lcd.display(img)

    print(img.get_histogram().get_statistics().l_mean())
    uart.writechar(img.get_histogram().get_statistics().l_mean())
    led_counter = led_counter - 1
    if (led_counter == 0):
        if (led_mode == 0):
            ledGreen.on()
        else:
            ledGreen.off()
        led_mode = 1 - led_mode
        led_counter = M_LED_COUNT
    while (clk + 100 > utime.ticks_ms()):
        pass

```

Используя Pioneer Station и OpenMV IDE, [загрузите](#) соответствующие программы на квадрокоптер и модуль камеры. Подключите аккумулятор к «Пионеру» и запустите выполнение программы. Протестируйте ее работу, направляя камеру на объекты с различной яркостью.

---

## 4.5 Модуль оптического позиционирования



Модуль позволяет позиционировать квадрокоптер, используя оптический датчик, который отслеживает смещение объектов. Это дает возможность удерживать положение и летать в автоматическом режиме без системы навигации и GPS. Модуль монтируется на плате расширения снизу и требует подключения шлейфом с базовой платой.

Модуль оснащен оптическим дальномером. Он может определять высоту над полом или препятствием в пределах 0-1,5 м. Если коптер поднимается выше этой отметки, зеленый светодиод на модуле перестает мигать и горит постоянно. Это говорит о том, что теперь для определения высоты используется барометр.

Подключите Пионер к Pioneer Station и активируйте кнопку OPT в параметрах автопилота, чтобы использовать модуль при полёте.

При программируемом полете используется команда `ap.goToLocalPoint(x, y, z)`. При запуске моторов квадрокоптер запоминает точку старта как (0, 0, 0). При этом ось y направлена вперед (по направлению квадрокоптера), ось x - вправо. Параметр z отвечает за высоту полёта. Значения (x, y, z) в метрах задают координаты точки, в которую полетит квадрокоптер, относительно точки старта.

---

**Примечание:** Чем больше суммарная протяженность полета, тем больше будет накопленная ошибка. При этом снижается точность позиционирования.

---

Модуль распознает поверхность, над которой летит квадрокоптер. Чем больше деталей (таких как орнамент или стыки между плитками) попадает в поле зрения, тем точнее будет полёт.

Управляя квадрокоптером с пульта, выберите режим работы с системой навигации (тумблер SwB в среднем положении), и «Пионер» будет использовать подключенный модуль оптического потока, что сделает полет более стабильным.

### 4.5.1 Обновление прошивки

Для обновления прошивки модуля на вашем компьютере должна быть установлена программа Pioneer Station. Нажмите и удерживайте кнопку «Старт» на базовой плате квадрокоптера, и подключите кабелем microUSB к компьютеру. Запустите Pioneer Station, выберите пункт меню «Обновление прошивки» и следуйте указаниям помощника. При переходе к пункту «Выбор устройства» в списке будет отображаться не только базовая плата квадрокоптера PioneerBase, но и модуль оптического потока ModuleOptFlow. Поставьте галочку напротив модуля и нажмите «Далее». При выборе источника прошивки рекомендуется «встроенный» - это последняя актуальная версия, входящая в сборку программы. Если версия модуля не определяется автоматически,

прочтите маркировку на обратной стороне платы и выберите соответствующую прошивку из доступных в папке Pioneer Station. Дождитесь окончания прошивки. После этого квадрокоптер перезагрузится в обычный режим.

## 4.5.2 Пример

Ниже приведен пример программы, использующий модуль оптического потока. Чтобы **загрузить программу на «Пионер»**, воспользуйтесь Pioneer Station. Включите Пионер, пульт, переведите переключатель SwB в нижнее положение и нажмите кнопку «Старт» на базовой плате квадрокоптера. Через 5 секунд квадрокоптер взлетит, пролетит на 1 метр вперед и вернется в исходную точку для посадки.

```
-- https://learnxinyminutes.com/docs/ru-ru/lua-ru/
--
local ledNumber = 4
--
local leds = Ledbar.new(ledNumber)
--
--                                table
local unpack = table.unpack

--
local curr_state = "PREPARE_FLIGHT"

--
--                                4- RGB
local function changeColor(color)
--
--                                0 3
  for i=0, ledNumber - 1, 1 do
    leds:set(i, unpack(color))
  end
end

--
--                                RGB                                changeColor
local colors = {
  {1, 0, 0}, --
  {1, 1, 1}, --
  {0, 1, 0}, --
  {1, 1, 0}, --
  {1, 0, 1}, --
  {0, 0, 1}, --
  {0, 0, 0} -- /
}

--
--                                ,
action = {
  ["PREPARE_FLIGHT"] = function(x)
    changeColor(colors[2]) --
    Timer.callLater(2, function () ap.push(Ev.MCE_PREFLIGHT) end) -- 2
    Timer.callLater(4, function () changeColor(colors[3]) end)-- 2 ( 4 ,
    Timer.callLater(6, function ()
      ap.push(Ev.MCE_TAKEOFF) -- 2 ( 6 )
      curr_state = "FLIGHT_TO_FIRST_POINT" --
    end)
  end,
  ["FLIGHT_TO_FIRST_POINT"] = function (x)
    changeColor(colors[4]) --
    Timer.callLater(2, function ()
      ap.goToLocalPoint(0, 1, 1) -- (0 , 1 , 1 )
      curr_state = "FLIGHT_TO_SECOND_POINT" --
    end)
  end,
end,
```

(continues on next page)

```

["FLIGHT_TO_SECOND_POINT"] = function (x)
  changeColor(colors[5]) --
  Timer.callLater(2, function ()
    ap.goToLocalPoint(0, 0, 0.8) -- (0 , 0 , 0.8 )
    curr_state = "PIONEER_LANDING" --
  end)
end,
["PIONEER_LANDING"] = function (x)
  changeColor(colors[6]) --
  Timer.callLater(2, function ()
    ap.push(Ev.MCE_LANDING) --
  end)
end
}

--
function callback(event)
  --
  if (event == Ev.TAKEOFF_COMPLETE) then
    action[curr_state]()
  end
  --
  if (event == Ev.SHOCK) then
    changeColor(colors[1])

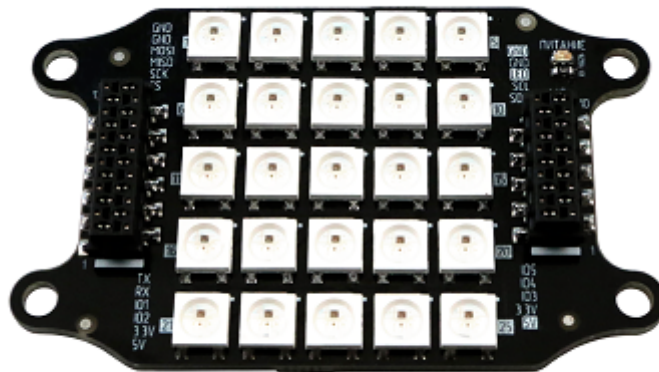
  end
  --
  if (event == Ev.POINT_REACHED) then
    action[curr_state]()
  end

  --
  if (event == Ev.COPTER_LANDED) then
    changeColor(colors[7])
  end
end

--
changeColor(colors[1])
--
Timer.callLater(2, function () action[curr_state]() end)

```

## 4.6 Модуль LED



Модуль LED - это плата с матрицей из 25 светодиодов. Модуль крепится к плате расширения снизу. Подключение через разъем X2 комплектным шлейфом.

Блок светодиодов может использоваться для подсветки или индикации событий в зависимости от заданной программы.

### 4.6.1 Пример

В качестве примера приведена программа, которая случайно меняет цвет блока светодиодов каждую секунду. Чтобы загрузить программу на «Пионер», воспользуйтесь Pioneer Station.

```
-- (4) + LED (25)
local ledNumber = 29
--
local leds = Ledbar.new(ledNumber)

-- , RGB
local function changeColor(red, green, blue)
  for i=0, ledNumber - 1, 1 do
    leds:set(i, red, green, blue)
  end
end

-- , timerRandomLED
local function emergency()
  timerRandomLED:stop()
  --
  Timer.callLater(1, function () changeColor(0, 0, 0) end)
end

function callback(event)
  --
  if (event == Ev.LOW_VOLTAGE2) then
    emergency()
  end
end

-- ,
timerRandomLED = Timer.new(1, function ()
  changeColor(math.random(), math.random(), math.random())
end)
```

(continues on next page)

```
--
timerRandomLED:start()
```

Чтобы освоить продвинутую работу с матрицей светодиодов, используйте пример ниже:

```
local led_count = 29 -- (4 + 25 )
local led_offset = 4 --
local matrix_count = 25 --
local leds = Ledbar.new(led_count) --
local unpack = table.unpack -- table
local colors = {
  red = {1, 0, 0}, -- RGB. 0 1
  green = {0, 1, 0},
  blue = {0, 0, 1},
  purple = {1, 0, 1},
  cyan = {0, 1, 1},
  yellow = {1, 1, 0},
  white = {1, 1, 1},
  black = {0, 0, 0} -- =
}

local dig = {
  {3, 7, 8, 13, 18, 22, 23, 24}, -- 1
  {2, 3, 4, 9, 12, 13, 14, 17, 22, 23, 24}, -- 2
  {2, 3, 4, 9, 12, 13, 14, 19, 22, 23, 24}, -- 3
  {2, 4, 7, 9, 12, 13, 14, 19, 24}, -- 4
  {2, 3, 4, 7, 12, 13, 14, 19, 22, 23, 24}, -- 5
  {3, 4, 7, 12, 13, 14, 17, 19, 22, 23, 24}, -- 6
  {2, 3, 4, 9, 13, 18, 23}, -- 7
  {2, 3, 4, 7, 9, 13, 17, 19, 22, 23, 24}, -- 8
  {2, 3, 4, 7, 9, 12, 13, 14, 19, 22, 23}, -- 9
  {1, 3, 4, 5, 6, 8, 10, 11, 13, 15, 16, 18, 20, 21, 23, 24, 25}, -- 10
  [0] = {2, 3, 4, 7, 9, 12, 14, 17, 19, 22, 23, 24} } -- Lua

↪ 1, 0

local ledMatrix = {} --

for i = 1, matrix_count + 1, 1 do
  ledMatrix[i] = colors.black --
end

--

local function updateMatrix()
  for i = led_offset, led_count - 1, 1 do
    leds:set(i, unpack(ledMatrix[i-led_offset + 1]))
  end
end

-- . x - ; y - ; colors - RGB
local function setPixelMatrix( x, y, colors )
  i = (y - 1) * 5 + x
  if ledMatrix [i] then
    ledMatrix [i] = colors
  end
end

-- . colors - RGB
local function fillMatrix( colors )
  for i = 1, matrix_count + 1, 1 do
    ledMatrix [i] = colors
  end
end

-- . x - ; colors - RGB
```

(continues on next page)

```
local function setDig( x, colors )
  for _, v in ipairs(dig[x]) do
    ledMatrix[v] = colors
  end
end
--

function callback( event )
end

-- . 0 9,
function digitOutput()
  colors_any[1], colors_any[2], colors_any[3] = fromHSV(col, 100, 10) --
  setDig( i, colors_any) --
  ↪ updateMatrix() --
  ↪ if col < 360 then --
    col = col + 1 --
  ↪ elseif i < #dig-1 then --
    fillMatrix( colors.black) --
  ↪ col = 0 --
  ↪ i = i + 1 --
  ↪ else --
    fillMatrix(colors.black) --
    col = 0 --
  ↪ i = 0 --
  end
  Timer.callLater(0.003, function () digitOutput() end) -- ,
  ↪
end

colors_any = {0,0,0} -- RGB
i = 0 --
col = 0 -- HSV
digitOutput() --
```



---

## 4.7 Подвес для камеры GoPro



Подвес крепится к нижней части рамы квадрокоптера винтами М3х5 и позволяет надежно зафиксировать GoPro для видеозаписи во время полета. Вертикальный угол наклона камеры регулируется с помощью затяжного винта.

При установленном модуле и камере соблюдайте особую осторожность при взлёте и посадке, т.к. подвес выступает за габариты защитного каркаса квадрокоптера и влияет на его баланс.

## 4.8 Камера для фото и видео съемки



Модуль подходит как для съемки видео, так и для полетов в режиме FPV. Камера Runcam Split закреплена на поворотном подвесе. В корпусе модуля спрятана плата с кнопками управления, разъемами и слотом, в который установлена карта microSD на 4 Гб. На ней по умолчанию сохраняются все фотографии и видеоролики.

Плата подключения дополнительных модулей не совместима с модулем камеры. Используйте стандартную крышку отсека аккумулятора. Снимите её с Пионера и притяните модуль камеры сверху четырьмя винтами. После этого снова установите крышку на Пионер так, чтобы модуль камеры оказался снизу. Снимите нижнюю крышку модуля, чтобы получить доступ к плате Runcam Split.

Подключите коннектор камеры к разъему X1 на базовой плате квадрокоптера. Для дальнейшей настройки подключите аккумулятор Пионера.

Камерой Runcam Split удобно управлять со смартфона по сети Wi-Fi. Для этого нужно скачать и установить приложение Runcam:

[Версия для IOS](#) | [Версия для Android](#)

Для управления камерой на плате есть 2 кнопки - «питание» (прямо напротив задней стенки камеры) и «переключение режимов» (там же, ближе к углу платы). Если до них сложно достать, попробуйте нажимать их отверткой из комплекта Пионера. Чтобы включить режим Wi-Fi, один раз нажмите на «переключение режимов». Должен прозвучать одиночный сигнал. Теперь камера работает как точка доступа. Включите Wi-Fi на смартфоне и подключитесь к сети RCSplit. Пароль 1234567890.

Запустите приложение Runcam. Выберите модель Split 2S и нажмите «Connect». Теперь изображение с камеры транслируется на экран смартфона, можно делать фотографии и управлять записью видео, настраивать параметры. Чтобы открыть альбом камеры и скопировать содержимое на смартфон, нажмите иконку слева от кнопки записи. Вы можете просматривать фото и видео на microSD-карте, удалять и копировать материалы прямо на microSD карте.

Если у вас остались вопросы по настройке камеры, скачайте [инструкцию к Runcam Split](#)

---

## 4.9 FPV комплект

FPV (First Person View - вид от первого лица).

Комплект позволяет оператору получать изображение с курсовой камеры с минимальной задержкой по времени. В комплект входит:

- камера Runcam Split
- передатчик видеосигнала с антенной
- FPV шлем с приемником

Как настроить:

Плата подключения дополнительных модулей не совместима с модулем FPV. Используйте стандартную крышку отсека аккумулятора. Снимите её с Пионера и притяните модуль FPV сверху четырьмя винтами. После этого снова установите крышку на Пионер так, чтобы камера оказалась снизу.

Подключите коннектор модуля к разъему X1 на базовой плате квадрокоптера. Снимите защитный колпачок с линзы камеры. Для дальнейшей настройки подключите аккумулятор Пионера.

Переднюю часть шлема можно отсоединить и использовать как монитор.

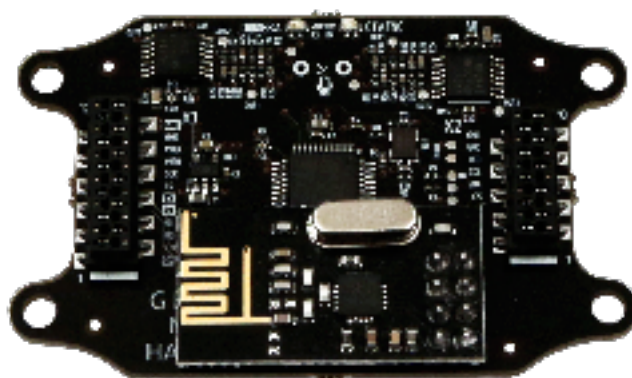
Подключите комплектную антенну к разъему на верхней панели шлема. Чтобы включить устройство, нажмите кнопку рядом с антенной. Кнопками Ch+ и Ch- переключайте каналы, пока на экране не появится изображение с камеры. Теперь вы можете управлять полетом Пионера, используя вид от первого лица.

---

**Примечание:** Используйте ползунок на плате видеопередатчика, чтобы выбрать один из 32-х каналов.

---

## 4.10 Бортовой модуль навигации в помещении



Модуль входит в комплект системы навигации в помещении вместе с блоком управления и четырьмя ультразвуковыми излучателями. Он монтируется на основной плате «Пионера» сверху с помощью 4-х винтов М3.

Модуль оснащен двумя микрофонами, которые позволяют контроллеру оценивать время прихода и разность фаз сигналов с излучателей. Далее происходит синхронизация с блоком управления по радиоканалу и определяется точное положение квадрокоптера в пространстве, а также его скорость.

Для работы модуля необходимо расположить включенный квадрокоптер в зоне действия ультразвуковых излучателей.

**Дополнительно:** *Система навигации в помещении Геоскан Локус*

### 4.10.1 Пример

Полет в системе навигации может осуществляться как в ручном режиме, так и по заранее загруженной программе. Пример такой программы - ниже. Выполняя её, «Пионер» взлетает, набирает высоту 1.2 м, затем летит в точку с координатами (1:1) и приземляется. Чтобы загрузить программу на «Пионер», используйте Pioneer Station.

```
--
--                                     table
local unpack = table.unpack

--
local ledNumber = 4
--
local leds = Ledbar.new(ledNumber)

--
local function changeColor(color)
  --                                     4-
  for i=0, ledNumber - 1, 1 do
    leds:set(i, unpack(color))
  end
end

--
--                                     RGB                                     changeColor
local colors = {
  {1, 0, 0}, --
  {0, 1, 0}, --
  {0, 0, 1}, --
  {1, 1, 0}, --
  {1, 0, 1}, --
  {0, 1, 1}, --
  {1, 1, 1}, --
  {0, 0, 0} -- /
}

--
--                                     {x,y,z}
local points = {
  {0, 0, 0.7},
  {0, 1, 0.7},
  {0.5, 1, 0.7},
  {0.5, 0, 0.7}
}

--
local curr_point = 1

--
local function nextPoint()
  --                                     . % -                                     , # -                                     .                                     ,
  --                                     ,                                     ,

```

(continues on next page)

```

curr_color = ((curr_point - 1) % (#colors - 2)) + 1
--
changeColor(colors[curr_color])
--
if(curr_point <= #points) then
  Timer.callLater(1, function()
    --
    ap.goToLocalPoint(unpack(points[curr_point]))
    --
    curr_point = curr_point + 1
  end)
--
else
  Timer.callLater(1, function()
    --
    ap.push(Ev.MCE_LANDING)
  end)
end
end

--
function callback(event)
  --
  Flight_com_takeoffAlt,
  if(event == Ev.TAKEOFF_COMPLETE) then
    nextPoint()
  end
  --
  if(event == Ev.POINT_REACHED) then
    nextPoint()
  end
  --
  if (event == Ev.COPTER_LANDED) then
    changeColor(colors[8])
  end
end

--
ap.push(Ev.MCE_PREFLIGHT)
--
changeColor(colors[7])
--
Timer.callLater(2, function() ap.push(Ev.MCE_TAKEOFF) end)

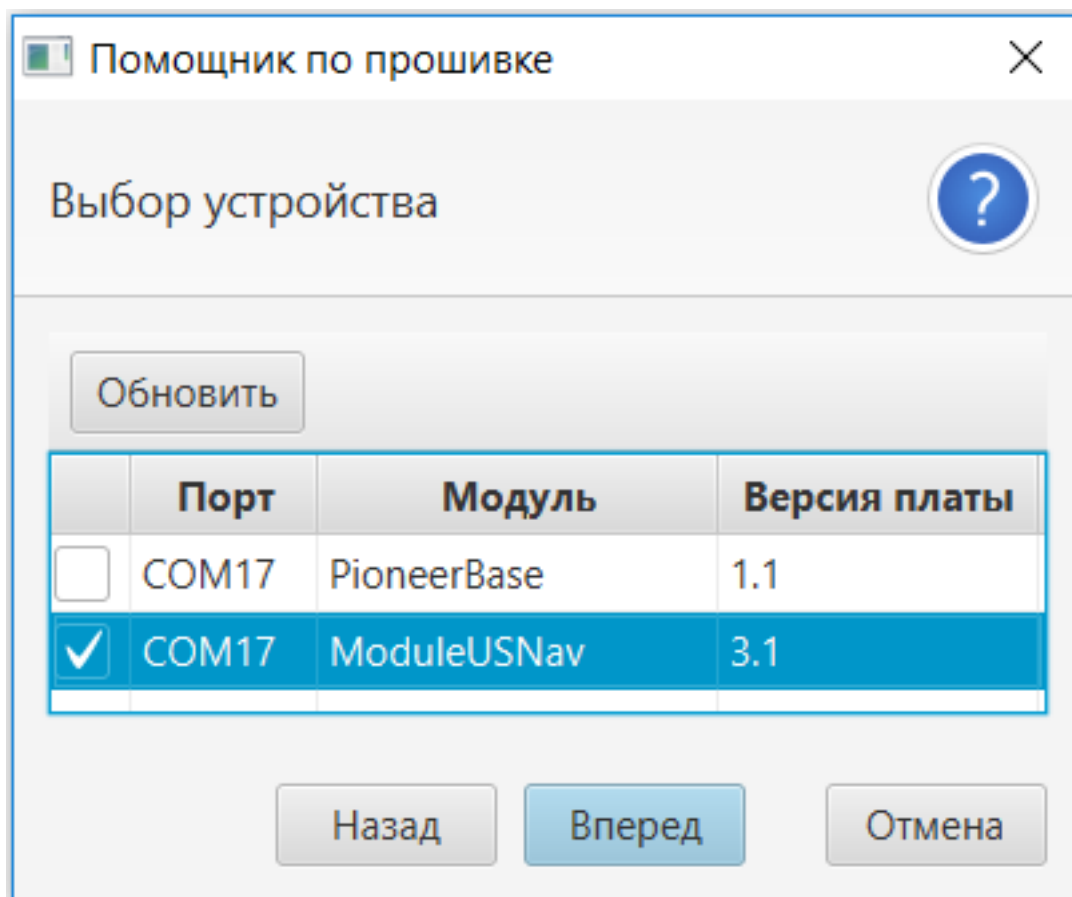
```

#### 4.10.2 Прошивка модуля навигации

Для обновления прошивки модуля на вашем компьютере должна быть установлена программа Pioneer Station. В окне программы выберите пункт меню «обновление прошивки» и следуйте указаниям помощника.

**Примечание:** Для перепрошивки необходимо подключить аккумулятор к базовой плате Пионера, удерживая нажатой кнопку «Старт». Без него на модуль не поступает питание и он не отображается в окне программы. В режиме перепрошивки на модуле должны гореть все светодиоды.

При переходе к пункту «Выбор устройства» в списке будет отображаться не только базовая плата квадрокоптера PioneerBase, но и модуль навигации ModuleUSNav. Поставьте галочку напротив модуля и нажмите «Далее».



При выборе источника прошивки рекомендуется «встроенный» - это последняя актуальная версия, входящая в сборку программы. Если версия модуля не определяется автоматически, прочтите маркировку на обратной стороне платы и выберите соответствующую прошивку из доступных в папке Pioneer Station. Дождитесь окончания прошивки. После этого квадрокоптер перезагрузится в обычный режим.

Кроме полета на дистанционном управлении, «Пионер» может следовать заложенной перед взлетом программе и автономно выполнять прописанные задачи. Для удобства программирования квадрокоптер поддерживает работу с такими инструментами как Pioneer Station и TRIK Studio, которые позволяют сформировать полетное задание и загрузить его в память «Пионера» при подключении к компьютеру. В этом разделе вы научитесь работать с квадрокоптером через интерфейс Pioneer Station, рассмотрите примеры полетных сценариев и узнаете как программировать подключаемые модули.

## 5.1 Визуальное программирование TRIK

# ТРИК

Графическая среда TRIK Studio позволяет составлять программы для роботов, используя наглядные функциональные блоки. Такой подход значительно упрощает работу с «Пионером». Даже не имея опыта программирования, вы сможете написать задание, загрузить его на квадрокоптер и осуществить запуск.

Чтобы установить программу, нажмите [скачать TRIK Studio](#) и запустите загруженный .exe файл

Для синхронизации с «Пионером» вам понадобится *Pioneer Station*.

### 5.1.1 Создание программы в TRIK Studio

При первом запуске программы создайте новый проект (кнопка в левом верхнем углу). Чтобы начать создавать программу, достаточно перетащить нужные блоки действия из «Палитры» на диаграмму поведения робота, а затем соединить их между собой.

#### Основы работы с TRIK Studio

- Каждая программа должна содержать блоки «Начало» и «Конец». В случае ветвления программы каждая ветвь должна либо вернуться в основное дерево, либо заканчиваться блоком «конец».

- Для соединения двух блоков наведите курсор мыши на первый, нажмите и удерживайте **правую** кнопку мыши, протяните линию до второго и отпустите. Появившаяся на диаграмме стрелка означает, что два блока логически связаны. Также можно выбрать нужный блок и захватить **левой** кнопкой мыши появившийся справа от него маркер, протянув до следующего блока и отпустив.
- Настроить выбранный блок можно в окне «Редактор свойств».
- Чтобы удалить ненужный блок, выберите его щелчком левой кнопки мыши и нажмите del.
- Если нужно выбрать сразу несколько блоков, их можно обвести рамкой, зажав левую кнопку мыши и потянув курсор. Также можно выделять блоки по одному, зажав ctrl.
- На диаграмме не должно оставаться красных стрелок и неподключенных блоков. Для удобства старайтесь выстраивать диаграмму в направлении слева направо и сверху вниз.
- Для удобства и скорости работы изучите список «горячих клавиш» в разделе «Настройки». Также полезно прочитать раздел **Помощь (F1)**

#### **Блоки, используемые для программирования «Пионера»:**

- **Условие** - позволяет создать два сценария действия «Пионера» в зависимости от заданного логического условия. У блока должны быть две исходящие связи, в одной из которых в редакторе свойств должно быть назначено значение параметра «условие» (истина или ложь)
- **Конец условия** - обозначает слияние двух веток условного оператора. Никаких действий не выполняет, но полезен для обеспечения структурности программы.
- **Инициализация переменной** - позволяет объявить новую переменную. В редакторе свойств или прямо на диаграмме задается имя переменной и ее значение.
- **Случайное число** - присваивает выбранной переменной случайное значение из выбранного диапазона.
- **Комментарий** - позволяет включить в программу текстовые пояснения, упрощающие понимание структуры участка или конкретного блока.
- **Таймер** - задает время ожидания перед выполнением следующего блока программы в миллисекундах.
- **Взлет, Посадка** - команды начала и завершения полета.
- **Лететь в точку** - указывает координаты точки, в которую полетит квадрокоптер. Координаты вводятся без точек и запятых в строки широты и долготы. Строка «высота» отображает расстояние (в метрах) от поверхности в конечной точке полета.
- **Лететь в точку(ЛК)** - команда аналогична предыдущей, но точка назначения задается в локальных координатах. За точку отсчета (0,0,0) принимается место взлета. Значения координат задаются в метрах.
- **Светодиод** - управляет работой светодиодов на плате «Пионера». Меняя значения для каждого цвета, можно зажигать светодиоды по одному или в различных комбинациях. В паре с блоком «Таймер» можно задать длительность свечения.
- **Магнит** - управляет работой модуля захвата груза. Чтобы включить магнит, поставьте галочку в чекбоксе значения свойств блока.
- **Команда** - позволяет выполнить написанную в свойствах команду. Для написания команды используется язык Lua. Для выполнения команды поставьте галочку в чекбоксе.



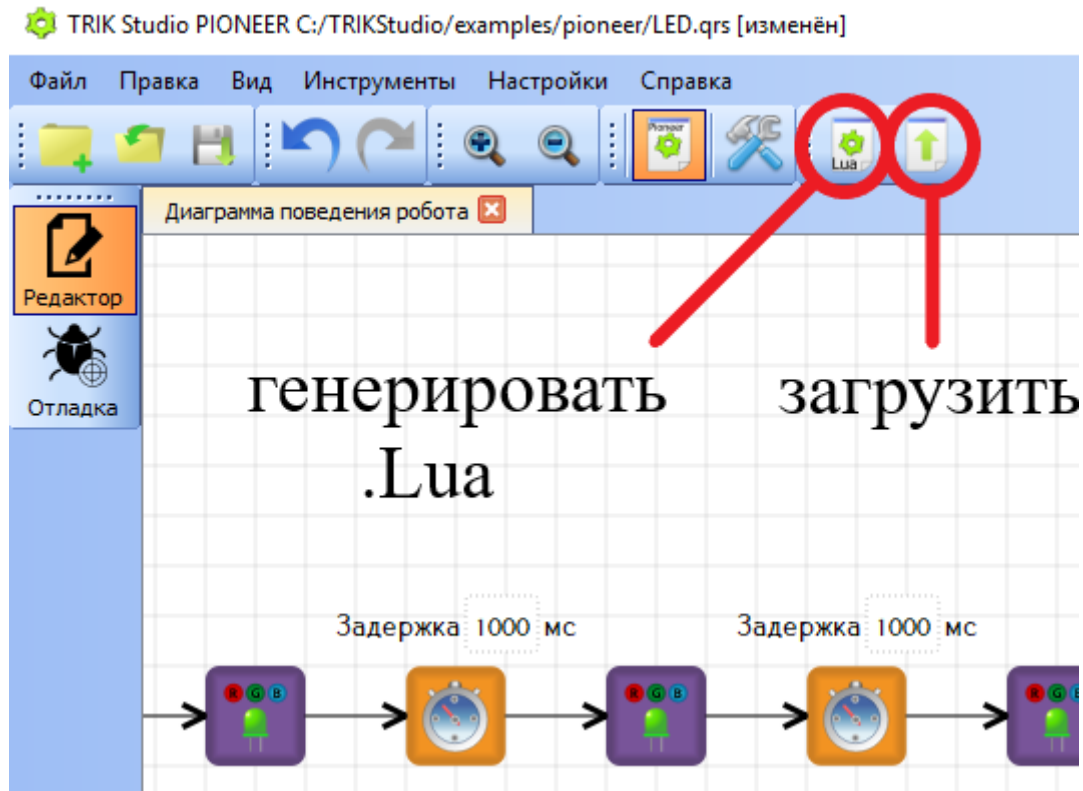
- **Рыскание** - управляет изменением направления полета «Пионера» вокруг вертикальной оси. Для поворота по часовой стрелке задайте значение угла в градусах со знаком «минус».

Комбинируя блоки между собой и создавая связи между ними, вы сможете создать любую полетную программу для «Пионера»

### 5.1.2 Пример программы

Для начала, а также в качестве примера, научимся программно управлять светодиодами на плате «Пионера». Для этого при запуске программы нажмите «Открыть проект» и выберите файл LED. В рабочей области появилась блок-схема программы, которая при запуске будет включать светодиоды на одну секунду каждым цветом. Вы можете внести изменения в любой блок программы или добавить новый, главное встроить его в структуру с помощью стрелок-соединений.

Диаграмма работы собрана, нужно сгенерировать соответствующий код в Lua. Для этого нажмите кнопку над рабочей областью, или воспользуйтесь сочетанием клавиш ctrl+shift+G. Кнопка «Загрузить программу на «Пионер»» понадобится нам позже.



Если все соединено без ошибок, на экране откроется вкладка с текстом программы. При обнаружении ошибок в нижней части экрана появится окно с пояснением что нужно исправить для корректной работы программы.

Код сгенерирован, теперь нажмите кнопку «Загрузить программу на «Пионер»». Программа откроется в редакторе Pioneer Station. Теперь осталось загрузить её в «Пионер».

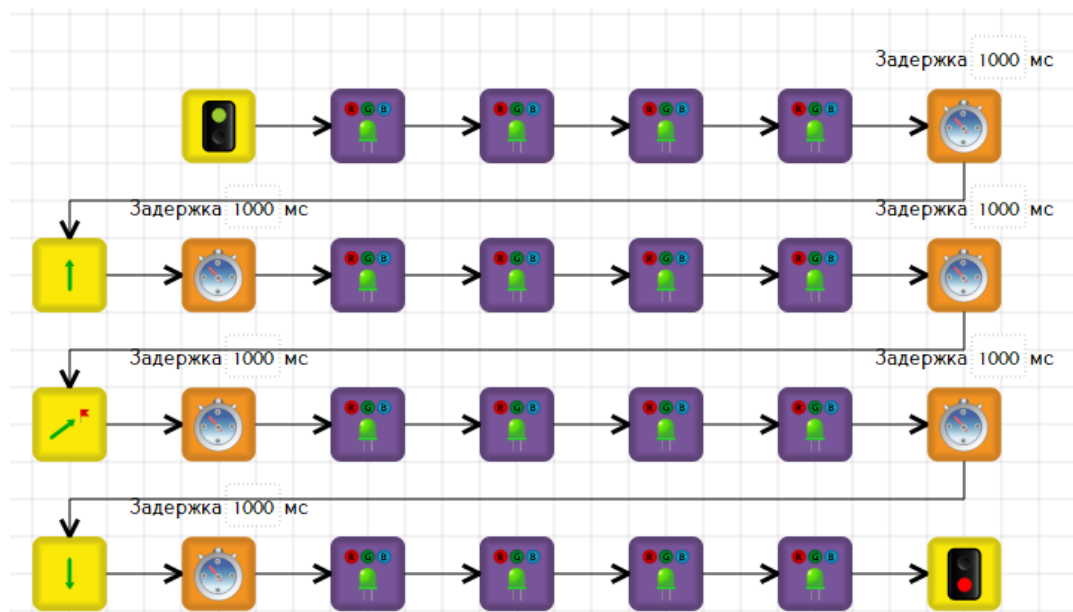
**Научившись работать со светодиодами, попробуйте более сложные варианты.**

### 5.1.3 Взлет-Миссия-Посадка

Для создания первой настоящей полетной программы можете взять за основу учебные примеры fly\_square или fly\_to\_point, предлагаемые при запуске TRIK Studio.

**Примечание:** Обратите внимание, что полет в точку осуществляется в локальной системе координат, которую обеспечивает работа *Система навигации в помещении Геоскан Локус*. Систему нужно настроить и активировать **перед взлетом**.

Для работы в географических координатах необходим *Модуль навигации GPS Глонасс*



Добавьте блоки «Рыскание» и «Светодиод», постарайтесь разнообразить условия вашей программы.

Построив диаграммы, не забудьте сгенерировать Lua код и проверить его на наличие ошибок. Загрузка программы в «Пионер» осуществляется так же как в предыдущем примере

## 5.2 Pioneer Station



Программа **Pioneer station** - это удобный инструмент работы с квадрокоптером. Она позволяет:

- Обновлять прошивку квадрокоптера
- Настроить параметры автопилота
- Разрабатывать и загружать полетные задания
- Запускать скрипты для тестирования

Чтобы установить программу, нажмите [скачать Pioneer station](#) и запустите загруженный .exe файл

---

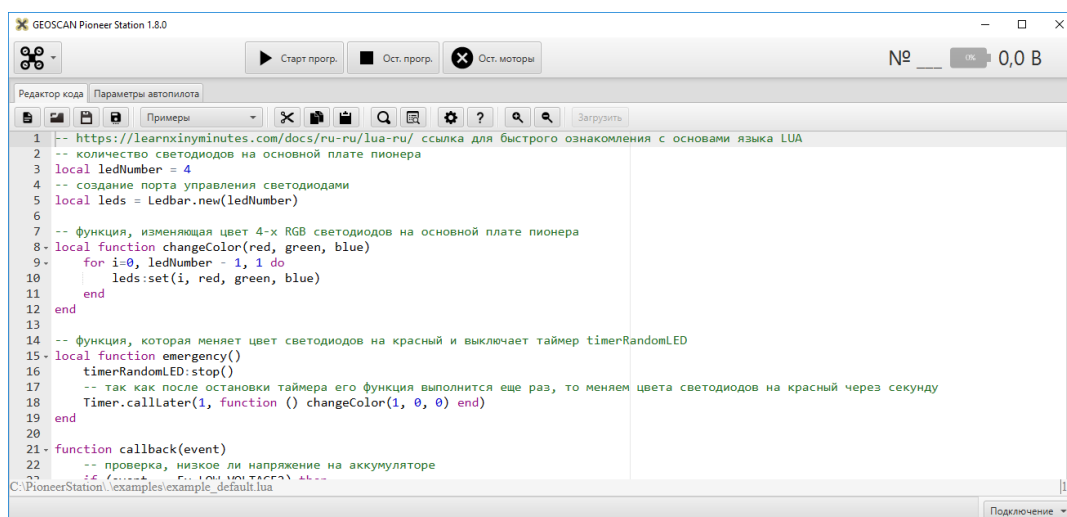
**Примечание:** Если программа не запускается попробуйте [скачать 32bit версию](#), распаковать архив и запустить run.bat.

---

Если вы используете ОС Linux, скачайте и распакуйте [архив PioneerStationLinux64](#)

## 5.2.1 Работа с Pioneer station

Интерфейс программы при запуске без подключенного «Пионера»:



В верхней части окна программы представлено **Главное меню**, в котором производится обновление **Pioneer station** и прошивки автопилота. Также в **Главном меню** производится настройка модема. В центральной части расположены кнопки запуска/остановки программы и остановки моторов. В правой части окна - индикатор подключения. После подключения в нем будет отображаться номер квадрокоптера и заряд аккумулятора. В центральной части окна располагаются окна: *Редактор кода* и *Настройка параметров автопилота*, работа с ними подробно описана в соответствующих разделах.

**Внимание:** Если ваша программа предполагает полет, не запускайте ее из интерфейса **Pioneer station**. Запуск квадрокоптера, подключенного по usb, опасен для вас и окружающих!

### Редактор кода





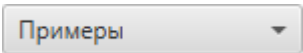










Этот раздел поможет научиться работать с Pioneer station

Основное окно **Pioneer station** - это редактор кода. В нём вы можете разрабатывать и редактировать полетные задания «Пионера» на языке Lua.

Главное меню редактора кода:



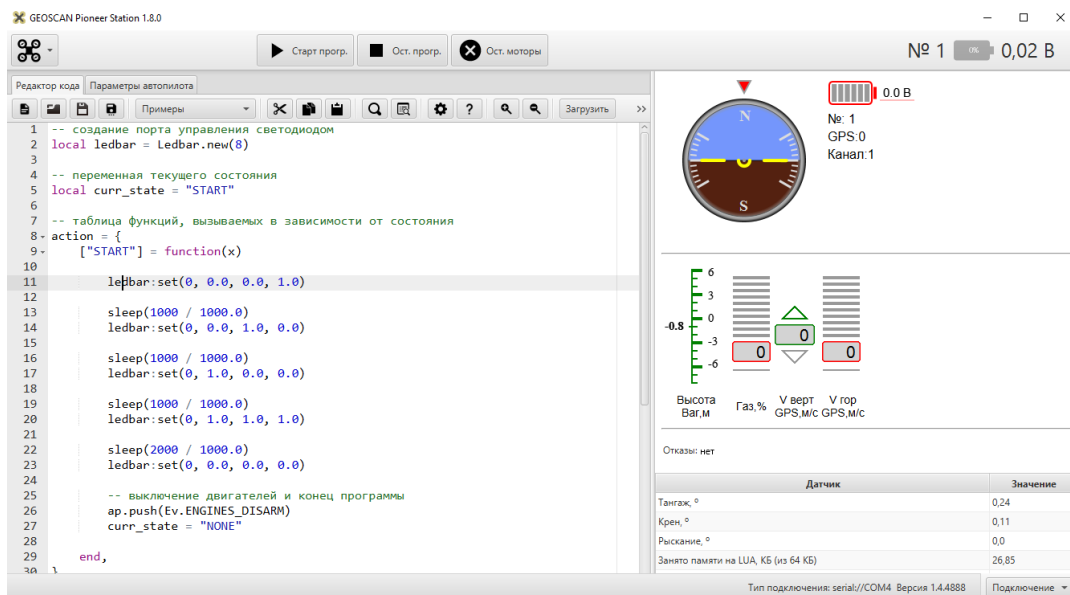
Назначение кнопок меню редактора:

- Создать новый файл 
- Открыть файл из папки 
- Сохранить файл 
- Сохранить как 
- Выбор одной из готовых программ примеров 
- Вырезать блок кода 
- Скопировать блок кода 
- Вставить блок кода 
- Поиск 
- Замена 
- Настройки редактора 
- Справка 
- Увеличить масштаб 
- Уменьшить масштаб 
- Загрузить программу на «Пионер» (активна только при подключении к квадрокоптеру) 

## 5.2.2 Загрузка программы в «Пионер»

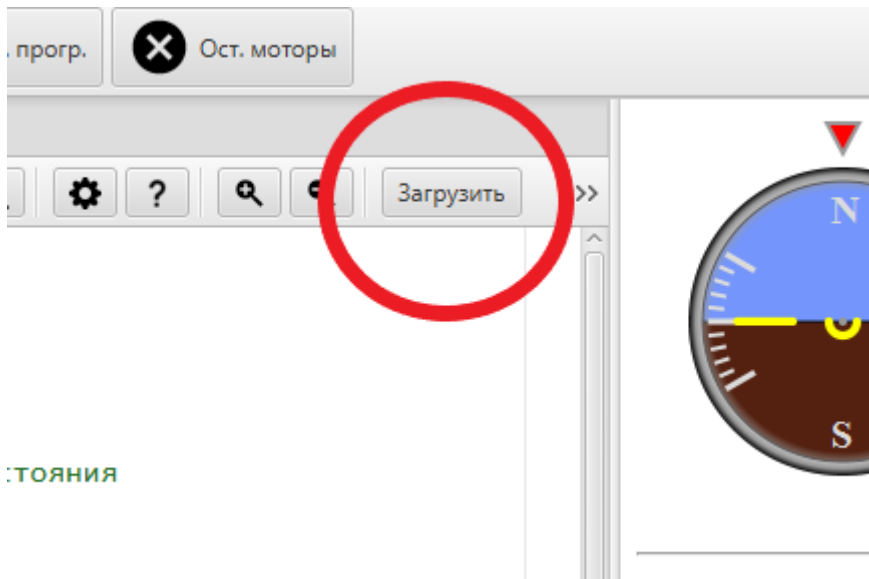
Имея код программы, созданный в TRIK Studio или написанный в Pioneer Station, можно приступить к его загрузке на квадрокоптер. Подключите «Пионер» к компьютеру USB кабелем и откройте нужную программу в Pioneer Station.

В правом нижнем углу Pioneer Station выберите вариант подключения «По кабелю USB». В окне программы должны появиться текущие параметры «Пионера», углы наклона, напряжение питания и номер.



**Примечание:** Попробуйте наклонять квадрокоптер в разные стороны и убедитесь, что авиагоризонт корректно отображает его положение. В случае ошибки соединения проверьте соединение кабеля и версию прошивки в Pioneer station, при необходимости обновите её.

Если программа изначально была создана в TRIK studio, нажмите кнопку «Загрузить программу на «Пионер»». Текст программы должен появиться в окне Pioneer Station. Теперь нажмите кнопку «Загрузить».



На плате «Пионера» должен несколько раз мигнуть светодиод «прием», а в программе - появиться уведомление об успешной загрузке.

Запустить выполнение программы можно разными способами. Если программа не предполагает полета, не отключая кабель от квадрокоптера нажмите кнопку «Старт прог.» в Pioneer Station, и через пять секунд сможете наблюдать результат её работы. Питания от USB достаточно для работы светодиодов и некоторых дополнительных модулей.

**При более сложных сценариях:** отключите квадрокоптер от компьютера, поме-

---

стите его в комнату или площадку, достаточно просторную для выполнения задач, и подключите аккумулятор. После звукового сигнала дождитесь, когда перестанут мигать основные светодиоды на плате, и нажмите кнопку «Старт» и отойдите от квадрокоптера. Через пять секунд начнется выполнение программы.

**Внимание:** Для безопасного полета, на случай нештатной ситуации держите в руках пульт, связанный с вашим «Пионером». Настройте на пульте режим автоматического полёта. Для этого переведите рычаг В в нижнее положение. Рычаги А, С, D – в верхнем положении. Для экстренного завершения программы переведите рычаг В в верхнее положение, управление квадрокоптером перейдет к вам.

### 5.2.3 Подключение по USB модему

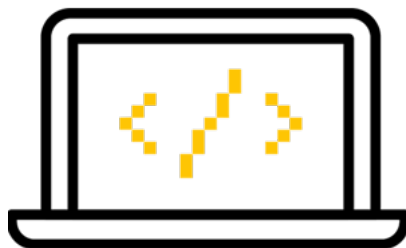
Для загрузки программ на «Пионер», помимо USB, может использоваться беспроводное соединение. Для подключения необходим радиомодем. Его можно подключить в любой свободный USB порт компьютера.



- Запустите Pioneer Station на компьютере, подключите «Пионер» кабелем USB и запомните номер канала, который появится в поле телеметрии справа от авиаризонта.
- В меню «Настройки» выберите пункт «Настроить модем». В открывшемся окне сохраните номер канала, который запомнили при выполнении предыдущего шага.
- Выберите квадрокоптер который хотите подключить по данному каналу и нажмите ОК.
- В дальнейшем для подключений «Пионера» к компьютеру после запуска Pioneer Station в меню «подключение» выбирайте вариант «По радиомодему». При этом на плату должно быть подано питание от основной батареи, либо от стороннего источника по кабелю USB.

---

## 5.3 Программирование на Lua



Стандартный набор «Пионера» позволяет программировать светодиоды и формировать полетное задание. В зависимости от уровня пользователя предлагается два варианта программирования «Пионера»:

Упрощенное визуальное программирование полетного задания «Пионера» в ПО [TRIK Studio](#)

Продвинутое программирование полетного задания «Пионера» на языке [Lua](#)

Также основы языка Lua приведены в файле `Learnlua.lua` в папке `pioneer station/examples`. Откройте его с помощью Pioneer Station и ознакомьтесь с синтаксисом и базовыми командами.

---

**Важно:** Работой дополнительных модулей также можно управлять программно через «Пионер». Изучите примеры в описании к модулям.

---

В разделе Lua - список и краткое описание команд, которые применяются для программирования «Пионера»

### 5.3.1 Lua документация по API

### 5.3.2 Описание API функций взаимодействия с автопилотом

`time()`

Возвращает время с момента включения коптера.

`deltaTime()`

Возвращает разницу в секундах между временем коптера, которое можно получить функцией `time()`, и глобальным временем системы навигации.

`launchTime()`

Возвращает время запуска для системы навигации.

`sleep(seconds)`

Останавливает выполнение скрипта на заданное время, допустимы дробные значения аргумента. **Рекомендуется использовать `Timer`, так как `sleep` блокирует дальнейшее выполнение скрипта.**

**Параметры** `seconds` – время сна в секундах.

`boardNumber`

Получение идентификационного номера борта - доступно через переменную.

**Пример**

```
local boardNumber = boardNumber
```

`ap.push(Event)`

Добавить событие автопилоту (см. *События, отправляемые автопилоту*).

**Параметры** `Event` – номер события или название (например, `Ev.COPTER_LANDED`).

### Пример

```
Ev.MCE_PREFLIGHT  
ap.push(Ev.MCE_PREFLIGHT)
```

`ap.goToPoint(latitude, longitude, altitude)`

Для полёта с использованием GPS.

### Параметры

- `latitude` – задается широта в градусах, умноженных на  $10^{-7}$ ;
- `longitude` – задается долгота в градусах, умноженных на  $10^{-7}$ ;
- `altitude` – задается высота в метрах.

Пример `ap.goToPoint(600859810, 304206500, 50)`

`ap.goToLocalPoint(x, y, z, time)`

Для полёта с использованием локальной системы координат.

### Параметры

- `x` – задается координата точки по оси *x*, в метрах;
- `y` – задается координата точки по оси *y*, в метрах;
- `z` – задается координата точки по оси *z*, в метрах;
- `time` – время, за которое коптер перейдет в следующую точку, в секундах. Если значение не указано, коптер стремится к точке с максимальной скоростью.

Пример `ap.goToLocalPoint(1, 1, 1.2)` или `ap.goToLocalPoint(1, 1, 1.2, 10)`

`ap.updateYaw(angle)`

Установить рыскание.

**Параметры** `angle` – угол в радианах.

## RGB светодиоды

```
class Ledbar
```

`new(Count)`

Создать новый `Ledbar` с заданным количеством светодиодов.

**Параметры** `Count` – количество светодиодов.

`set(self, num, r, g, b)`

Установить цвет на конкретный светодиод.

### Параметры

- `num` – номер светодиода, нумеруются с 0 по 3 на плате, далее последовательно по подключенным модулям;
- `r` – интенсивность красной компоненты цвета в интервале от [0;1];
- `g` – интенсивность зеленой компоненты цвета в интервале от [0;1];
- `b` – интенсивность синей компоненты цвета в интервале от [0;1].



Смотри также `fromHSV()`

### Пример

К коптеру дополнительно подключен модуль с магнитом, на котором 4 RGB светодиода.

```
local leds = Ledbar.new(8)
for i = 0, 3, 1 do
    leds:set(i, 1, 0, 0)
end

for i = 4, 7, 1 do
    leds:set(i, 0, 0.5, 0)
end
```

`fromHSV(hue, saturation, value)`

Конвертирует представление цвета из HSV в RGB. Можно использовать для задания цвета светодиода.

#### Параметры

- `hue` – задает цветовой тон. Варьируется в пределах [0;360];
- `saturation` – задает насыщенность. Варьируется в пределах [0;100];
- `value` – задает значение цвета. Варьируется в пределах [0;100];

**Результат** Возвращает три компоненты цвета r, g, b.

## GPIO

`class Gpio`

`new(Port, Pin, Mode)`

Создать GPIO на порте с настройками.

#### Параметры

- `Port` – Gpio.A; Gpio.B; ... Gpio.E;
- `Pin` – номер пина на порте;
- `Mode` – Gpio.INPUT, Gpio.Output, Gpio.ALTFU.

`read(self)`

Получить значение.

`set(self)`

Установить значение в 1.

`reset(self)`

Установить значение в 0.

`write(self, value)`

**Параметры** `value` – установить значение.

`setFunction(self, num)`

Задать номер альтернативной функции.

### Пример

```
local pin_name = Gpio.new(Gpio.A, 1, Gpio.OUTPUT)
pin_name:read() --
pin_name:set() --          1
pin_name:reset() --      0
pin_name:write(true) --   true
pin_name:setFunction(1) --
```

## UART

```
class Uart
```

```
new(num, rate, parity, stopBits)
```

Создать Uart на порте с настройками.

### Параметры

- *num* – номер UART;
- *rate* – скорость;
- *parity* – Uart.PARITY\_NONE, Uart.PARITY\_EVEN, Uart.PARITY\_ODD, необязательный параметр, по умолчанию Uart.PARITY\_NONE;
- *stopBits* – Uart.ONE\_STOP, Uart.TWO\_STOP, необязательный параметр, по умолчанию Uart.ONE\_STOP.

```
read(self, size)
```

Прочитать *size* байт.

```
write(self, data, size)
```

Записать данные (*data*) длиной (*size*).

```
bytesToRead(self)
```

Количество данных доступных для чтения.

```
setBaudRate(self, rate)
```

Установить скорость.

**Параметры** *rate* – скорость uart.

### Пример

```
local uart = Uart.new(1, 115200)
uart:read(10) --      10
```

## SPI

```
class Spi
```

```
new(num, rate, seq, mode)
```

Создать Spi на порте с настройками.

### Параметры

- *num* – номер Spi
- *rate* – скорость
- *seq* – Spi.MSB, Spi.LSB, Spi.MSB\_16, Spi.LSB\_16, необязательный параметр, по умолчанию Spi.MSB;
- *mode* – Spi.MODE0, Spi.MODE1, Spi.MODE2, Spi.MODE3, необязательный параметр, по умолчанию Spi.MODE0.

```
read(self, size)
```

Прочитать *size* байт.

```
write(self, data, size)
```

Записать данные (*data*) длиной (*size*).

```
exchange(self, data, size)
```

Записать данные (*data*) длиной (*size*) и прочитать *size*.

### Пример

```
local spi = Spi.new(2, 1000000)
spi:exchange("hello", 5) --      (data)      (size)      size
```

---

## Таймеры

class Timer

`new(sec, func)`

Создать новый Timer.

### Параметры

- `sec` – время интервала в секундах;
- `func` – функция, которая будет вызываться с заданным интервалом.

`start(self)`

Запускает таймер.

`stop(self)`

Останавливает таймер. При этом остановка уже запущенного таймера произойдет после выполнения функции, стоящей в очереди на выполнение.

`callAt(local_time, func)`

Создает и запускает новый Timer с функцией, которая будет вызвана один раз.

### Параметры

- `local_time` – локальное время (возвращаемое функцией `time()`), указывающее момент вызова функции;
- `func` – функция, которая будет вызвана.

`callLater(delay, func)`

Создает и запускает новый Timer с функцией, которая будет вызвана один раз.

### Параметры

- `delay` – время, через которое будет вызвана функция;
- `func` – функция, которая будет вызвана.

`callAtGlobal(global_time, func)`

Создает и запускает новый Timer с функцией, которая будет вызвана один раз.

### Параметры

- `global_time` – глобальное время (`time() + deltaTime()`), указывающее момент вызова функции;
- `func` – функция, которая будет вызвана.

---

**Примечание:** При использовании функций `callAt()`, `callLater()`, `callAtGlobal()` следует обратить внимание, что может быть не более 16 одновременно ожидающих таймеров. Если количество одновременно ожидающих таймеров больше 16, то новый таймер не будет создан.

---

## Пример

Смотри *Пример скрипта*

## События, отправляемые автопилоту

События (events) представляются константами с префиксом «Ev.»

Название	Описание
MCE_PREFLIGHT	Запустить двигатели и провести подготовку
ENGINES_DISARM	Отключить двигатели
MCE_LANDING	Отправить на посадку
MCE_TAKEOFF	Отправить на взлет
–устаревшие–	
ENGINES_ARM	Завести двигатели

---

### 5.3.3 Получение данных от автопилота

Для получение данных от автопилота используется класс Sensors

```
class Sensors
```

```
    lpsPosition()
```

**Результат** x, y, z

```
    lpsVelocity()
```

**Результат** vx, vy, vz

```
    lpsYaw()
```

**Результат** yaw

```
    orientation()
```

Данные положения.

**Результат** roll, pitch, azimuth

```
    altitude()
```

Данные высоты по барометру.

**Результат** высота в метрах

```
    range()
```

Данные с датчиков расстояния.

**Результат** Возвращает значения с датчика расстояния. Возвращает несколько значений.

```
    accel()
```

Данные с акселерометра.

**Результат** ax, ay, az

```
    gyro()
```

Данные с гироскопа.

**Результат** gx, gy, gz

```
    rc()
```

Данные с пульта управления.

**Результат** channel1, channel2, channel3, channel4, channel5, channel6, channel7, channel8

#### Примеры

```
local lpsPosition = Sensors.lpsPosition
local lpsVelocity = Sensors.lpsVelocity
local lpsYaw = Sensors.lpsYaw
local orientation = Sensors.orientation
local range = Sensors.range
local accel = Sensors.accel
local gyro = Sensors.gyro
local rc = Sensors.rc

lpsX, lpsY, lpsZ = lpsPosition()
lpsVelX, lpsVelY, lpsVelZ = lpsVelocity()
yaw = lpsYaw()

roll, pitch, azimuth = orientation()

range1, range2, _, _, range3 = range()

ax, ay, az = accel()
gx, gy, gz = gyro()
aileron, _, _, _, _, _, ch8, = rc()
```

### 5.3.4 Описание необходимых служебных функций скрипта

```
function callback(event) --
end
```

Доступны следующие события, приходящие от автопилота:

Название	Описание
ENGINES_STARTED	Двигатели запущены
COPTER_LANDED	Коптер совершил посадку
TAKEOFF_COMPLETE	Коптер достиг высоты взлета
POINT_REACHED	Коптер достиг точки
POINT_DECELERATION	Коптер начал тормозить при подлёте к точке
LOW_VOLTAGE1	Низкое напряжение аккумулятора, для возвращения домой
LOW_VOLTAGE2	Низкое напряжение аккумулятора, переходит в режим посадки
SYNC_START	Получен сигнал синхронного старта от системы навигации
SHOCK	Столкновение или слишком сильные вибрации
CONTROL_FAIL	Угол наклона коптера превысил допустимый
ENGINE_FAIL	Отказ двигателя

- событие Ev.ALTITUDE\_REACHED (коптер достиг высоты взлёта) начиная с версии АП 1.5.6173 более не используется.

### 5.3.5 Пример скрипта

```
local boardNumber = boardNumber
local unpack = table.unpack
local points = {
    {-0.6, 0.3, 0.2},
    {0.6, 0.3, 0.2},
    {0, 0, 0.5},
    {0.6, -0.3, 0.2}
}

local curr_point = 1

local function nextPoint()
    if(#points >= curr_point) then
        ap.goToLocalPoint(unpack(points[curr_point]))
        curr_point = curr_point + 1
    else
        ap.push(Ev.MCE_LANDING)
    end
end

function callback(event)
    if(event == Ev.TAKEOFF_COMPLETE) then
        nextPoint()
    end
    if(event == Ev.POINT_REACHED) then
        nextPoint()
    end
end

local leds = Ledbar.new(1)
local blink = 0
```

(continues on next page)

(продолжение с предыдущей страницы)

```
leds:set(0,1,1,1)
timerBlink = Timer.new(1, function ()
    if(blink == 1) then
        blink = 0
    else
        blink = 1
    end
    leds:set(0, blink, 0, 0)
end)
timerBlink:start()
ap.push(Ev.MCE_PREFLIGHT)
Timer.callLater(1, function() ap.push(Ev.MCE_TAKEOFF) end)
```

### 5.3.6 Описание пинов разъемов модулей

Пины МК АП на Pioneer\_Base\_v.1.0-v.1.1, выведенные на внешние разъемы:

Разъем X1 (пин МК)	Функция	Описание
1	Питание 5 В (только с АКБ)	Максимум 2 А
2	Питание 3.3 В	Максимум 2 А
3 (PA12)	USART1_RTS	
4 (PA11)	USART1_CTS, TIM1_CH4	Module_OpenMV
5 (PA10)	USART1_RX, TIM1_CH3	Module_GPS, Module_USNav
6 (PA9)	USART1_TX, TIM1_CH2	Module_GPS, Module_USNav
7 (PA15)	SPI3_NSS, TIM2_CH1	Module_GPS, Module_OpenMV
8 (PC10)	SPI3_SCK	Module_GPS, Module_OpenMV
9 (PC11)	SPI3_MISO	Module_GPS, Module_OpenMV
10 (PB5)	SPI3_MOSI, TIM3_CH2	Module_GPS, Module_OpenMV
11	Земля	
12	Земля	

Разъем X2 (пин МК)	Функция	Описание
1	Питание 5 В (только с АКБ)	Максимум 2 А
2	Питание 3.3 В	Максимум 2 А
3 (PC2)	ADCx_IN12	
4 (PC3)	ADCx_IN13	
5 (PA1)	ADCx_IN1, TIM2_CH2, TIM5_CH2	Module_Cargo (упр. магнитом)
6 (PB7)	I2C1_SDA, TIM4_CH2	Module_ToF, Module_OpenMV
7 (PB6)	I2C1_SCL, TIM4_CH1	Module_ToF, Module_OpenMV
8 (PA0)	DATA WS2812B	Уровень 5 В. Module_LED
9	Земля	
10	Земля	

Для Pioneer\_Base\_v.1.2:

Разъем X2 (пин МК)	Функция	Описание
3 (PA0)	USART4_TX, ADCx_IN0, TIM2_CH1, TIM5_CH1	Module_OpenMV
4 (PA1)	USART4_RX, ADCx_IN1, TIM2_CH2, TIM5_CH2	Module_OpenMV
5 (PC3)	ADCx_IN13, SPI2_MOSI	Module_Cargo (упр. магнитом)
8 (PC12)	DATA WS2812B	Уровень 5 В. Module_LED



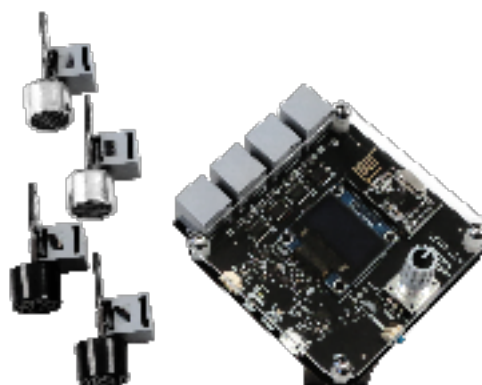


---

### Система навигации в помещении Геоскан Локус

---

Система позволяет создать контролируемую полетную зону с максимальным размером 10х10х4 метров. Она обеспечивает точное и безопасное управление квадрокоптером независимо от других систем навигации (GPS/ГЛОНАСС)



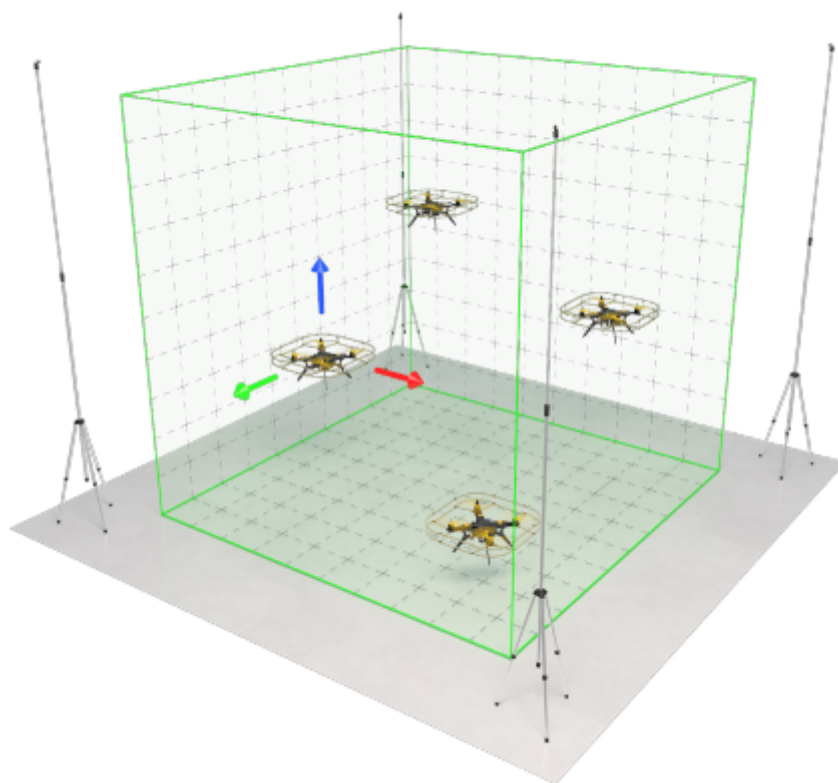
[Скачать pdf инструкцию по настройке системы навигации](#)

Комплект системы:

- блок управления;
- 4 ультразвуковых излучателя;
- бортовой модуль;
- соединительные провода;
- программа LPS для синхронизации с компьютером ([скачать последнюю версию](#));

---

## 6.1 Установка полетной зоны



Установите ультразвуковые излучатели в будущей зоне полета. Расположите их в вершинах полетной зоны так, чтобы они были направлены к ее центру. При этом:

- минимальная высота установки излучателей - 2 м
- минимальная дистанция между излучателями - 3 м

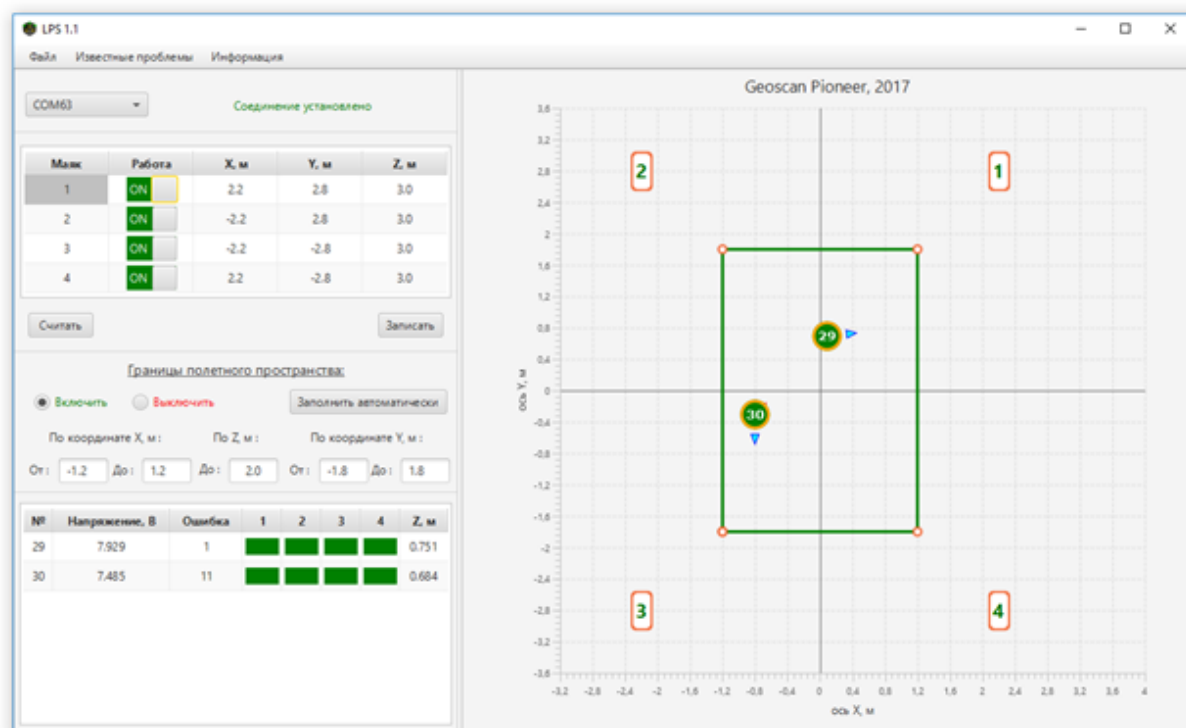
---

**Примечание:** Система будет работать точнее, если на полу в помещении будет мягкое покрытие (ковер, ковровин).

---

Проводами соедините каждый излучатель с блоком управления. Следите, чтобы не перепутать провода и разъемы, они подписаны. Блок управления лучше расположить за пределами полетной зоны. Его можно подключить к ноутбуку или компьютеру через порт USB. Блок управления и бортовой модуль общаются по радиоканалу.

Рулеткой или дальномером измерьте расстояния между излучателями и высоту их установки. Удобнее всего задавать полетную зону в форме квадрата или прямоугольника. По результатам измерений формируется координатная сетка полетной зоны.



На рисунке показан интерфейс программы LPS, которая используется для настройки и работы с системой позиционирования. При первом запуске программы заполните пустые поля координат для маяков. Как это сделать:

- Расстояние (в метрах) между излучателем 1 и 2 поделите на 2. Полученное значение введите в поле X со знаком «минус» для маяков №2 и №3 и без знака «минус» для маяка №1 и 4.
- Расстояние между излучателем 2 и 3 также поделите на 2 и введите значение в поле Y со знаком «минус» для маяков №3 и №4 и без знака «минус» для маяков №1 и №2.
- В поле Z введите высоту каждого излучателя над уровнем пола.

После ввода координат полетная зона сформирована и отображается в правом окне программы зеленой линией. По умолчанию углы полетной зоны отстоят от излучателя на 1 м.

**Примечание:** Параметры полетной зоны также можно задать напрямую на блоке управления

Нажмите кнопку на плате управления, чтобы включить систему. После этого должен загореться белый светодиод «питание» и зеленый «статус». Если этого не произошло, проверьте заряд аккумулятора.

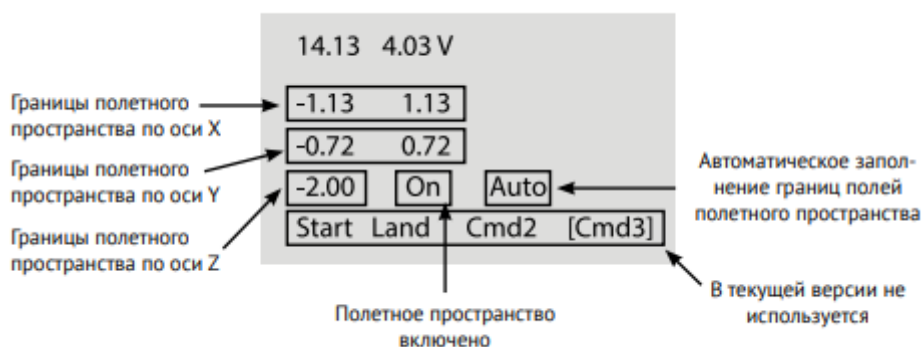
Вверху встроенного экрана отображается время с момента включения и текущий уровень напряжения аккумулятора. При падении напряжения ниже 3 В начнет мигать светодиод «статус», сообщая о необходимости подзаряда. В этом случае подключите плату через порт micro-usb к компьютеру или зарядному устройству (output 5В 2А).

На встроенном экране координаты отображаются так же, как в интерфейсе программы LPS.



Чтобы изменить любой из параметров системы, выберите его поворотом селектора, а затем нажмите на него. После коррекции (также поворотом) нажмите на селектор второй раз, чтобы подтвердить изменения и вернуться в меню.

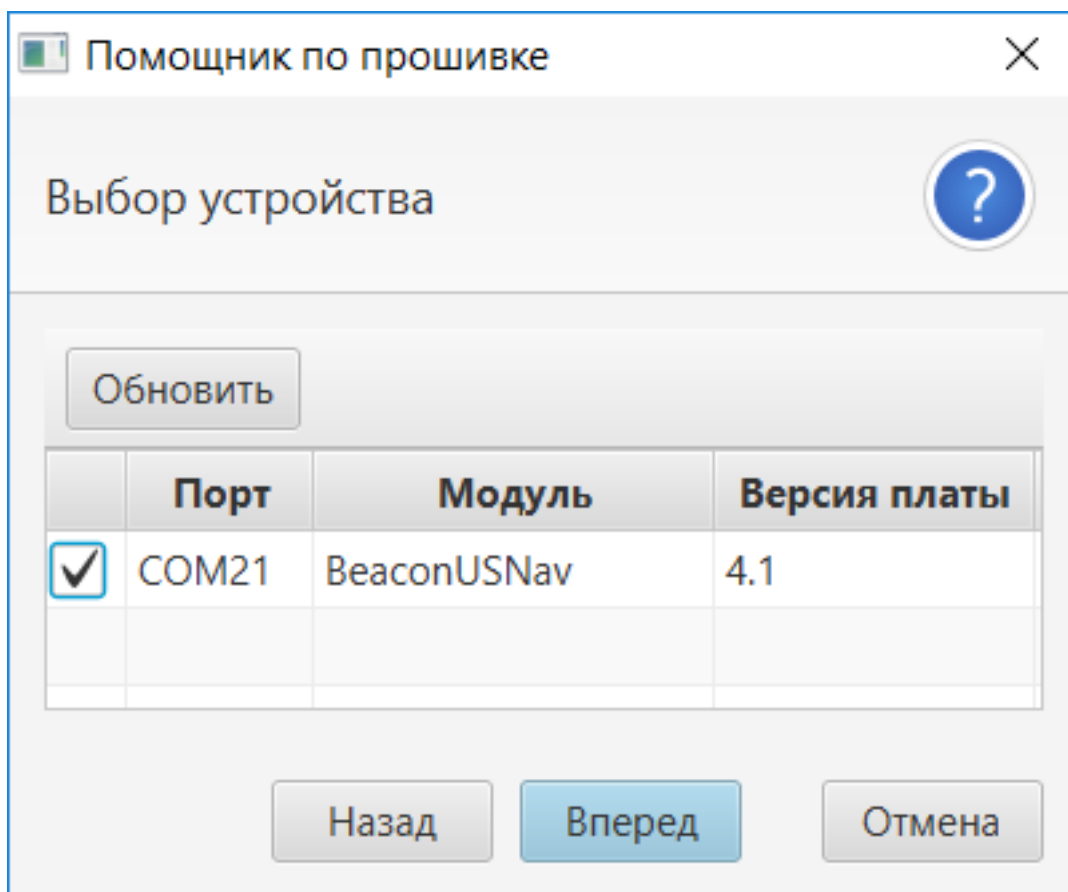
На втором экране меню можно вручную настроить отступы от границы полетного пространства или включить режим «Auto» - тогда все настроится автоматически. Там же можно отключать ограничения полетной зоны.



**Примечание:** Для работы системы подключение к компьютеру необязательно. Достаточно включить блок управления и расположить «Пионер» с установленным бортовым модулем в пределах полетной зоны.

## 6.2 Обновление прошивки системы навигации

Чтобы обновить прошивку, вам понадобится компьютер с установленной программой Pioneer Station. Запустите её, выберите в меню пункт «обновление прошивки» и следуйте указаниям помощника. Нажмите и удерживайте кнопку «меню» на плате, и не отпуская её включите блок управления. Так он перейдет в режим обновления прошивки. Теперь подключите блок управления системы навигации к компьютеру кабелем USB. При выборе устройства, поставьте галочку напротив модуля VeaconUSNav.



Рекомендуется выбирать встроенную прошивку, так вы установите последнюю стабильную версию. Процесс может занять до 5 минут, не отключайте блок управления до окончания прошивки.



**Если при попытке взлета квадрокоптер заваливается в сторону** - Проверьте правильность установки моторов. Моторы должны быть расположены в соответствии с указаниями на раме. Белые и черные моторы располагаются по диагонали относительно друг друга.

**Если при увеличении газа квадрокоптер крутит винтами, но не взлетает** - Проверьте правильность установки воздушных винтов. Воздушные винты должны быть расположены в соответствии с указаниями на раме. Надписи на винтах должны быть сверху.

**Если при инициализации один или несколько моторов не крутятся** - Необходимо проверить надежность соединения моторов с платой.

**Если при запуске винтов с пульта квадрокоптер никак не реагирует** - Проверьте правильность настройки пульта и привяжите приемник к пульту еще раз.

**Если квадрокоптер плохо управляется** - Необходимо проверить заряд аккумуляторных батарей квадрокоптера и пульта дистанционного управления.

**Если при подключении аккумулятора квадрокоптер никак не реагирует** - Проверьте целостность аккумулятора и уровень напряжения, он должен быть не ниже 6,6 В.

**Если в полете квадрокоптер сильно вибрирует** - Необходимо проверить правильность сборки рамы, защиты и крепления базовой платы. Также возможной причиной может быть дефект пропеллера.

**Если при подключении к Pioneer Station список портов пуст** - проверьте USB кабель; скачайте и установите [драйвер виртуального порта](#) для вашей операционной системы.

**Пустое окно при открытии вкладки параметров из файла в Pioneer Station** - Закройте программу и запустите файл run\_extra.bat из папки, в которой установлена Pioneer Station.

**Если коптер ведёт себя нестабильно при полёте по программе** - в параметрах автопилота установите значение `Copter_pos_aMax=1` и сохраните изменения. Этот параметр отвечает за плавность смены направления и скорости полёта квадрокоптера.

**Если моторы начинают крутиться с большой задержкой после команды на запуск, и коптер неверно реагирует на движение стиков управления** - в меню пульта **Syst** выберите пункт **Stick.Adj** и двигайте стики так, чтобы они достигли всех крайних положений (в том числе диагональных). После этого нажмите **назад** и попробуйте выполнить полёт снова.





### 8.1 20/06/2019

1. Обновлена Pioneer Station (Версия 1.10.0)
  - Добавлена поддержка нового протокола (1.5) для работы с новой версией автопилота
  - Добавлена возможность скачивать логи двумя способами:
    - Через Payload (старый медленный вариант), требуется бутлоадер с поддержкой возможности скачать лог
    - Через Plazlink (новый быстрый метод), требует лишь наличия установленной прошивки с версией протокола 1.5
  - Обновлены примеры скриптов LUA для программирования Пионера, оптимизирована структура и комментарии
  - Добавлены новые более сложные примеры для составления полетного задания и работы с модулем LED
  - Добавлена возможность просматривать логи с автопилота (в МЕНЮ в разделе “Работа с лог-гами”)
2. Обновлена прошивка автопилота (Версия 1.5.6173)
  - Обновлен протокол до версии 1.5
  - Добавлена возможность скачивать лог
  - Обновлен алгоритм взлета
  - Исправлен баг с генерацией 2-х событий во время взлета, теперь генерируется только одно - **Ev.TAKEOFF\_COMPLETE**. Событие **Ev.ALTITUDE\_REACHED** более не используется
  - Добавлена проверка старта двигателей при полете по программе

### 8.2 11/02/2019

1. Обновлена Pioneer Station (Версия 1.9.1)
  - Исправлено отображение кода и телеметрии при изменении окна программы
  - Исправлено отображение версии автопилота в таблице

- 
- Исправлено уведомление об обновлении прошивки модуля навигации в помещении
  - Добавлена ссылка на Changelog в меню-документация

### 8.3 1/02/2019

#### 1. Обновлена Pioneer Station(Версия 1.9.0)

- Добавлена калибровка акселерометра
- Добавлена калибровка компаса
- Добавлен вывод версии прошивки модуля, установленного на Пионер. Если прошивка устарела, то в таблице ее номер подсвечен и выскакивает окно с предложением обновить.
- Локализация на английский язык, теперь при установке приложения будет возможность выбрать между русским и английским.
- Исправлена ошибка с переключениями между режимами полетов по системе навигации, по GPS или по оптическому позиционированию
- Добавлено решение проблемы с пустым окном сравнения параметров.

#### 2. Обновлена прошивка автопилота (Версия 1.4.4922)

- Доработан алгоритм удержания курса.
- Исправлена ошибка с перезагрузкой автопилота при запуске нескольких таймеров.
- В Lua изменен вывод данных с барометра, теперь выводятся не сырые значения, а отфильтрованные автопилотом - нужно для заданий в соревнованиях Робофест.

#### 3. Обновлена прошивка модуля системы навигации (Версия 2.1)

- Доработан алгоритм захвата данных от излучателей