

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б10-мм

*Карасева Елизавета Олеговна*

# Загрузка программы в квадрокоптер Геоскан Пионер Мини из TRIK Studio

Отчёт по учебной практике  
в форме «Производственное задание»

Научный руководитель:  
старший преподаватель кафедры СП, Я.А. Кириленко

Санкт-Петербург  
2023

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Постановка задачи</b>	<b>4</b>
<b>2. Обзор</b>	<b>5</b>
2.1. TRIK Studio и Pioneer Station . . . . .	5
2.2. Pioneer SDK (Software Development Kit) . . . . .	7
2.3. Используемые технологии . . . . .	8
<b>3. Реализация</b>	<b>10</b>
3.1. Подготовка и настройка окружения . . . . .	10
3.2. Выбор способа реализации . . . . .	11
3.3. Реализация инструмента TRIK Studio Pioneer Tool . . . .	12
<b>4. Дополнения к функциональности</b>	<b>15</b>
4.1. Тестирование . . . . .	15
4.2. Оформление . . . . .	15
<b>Заключение</b>	<b>17</b>
<b>Список литературы</b>	<b>18</b>

# Введение

Робототехника, в частности, управление квадрокоптерами — постепенно становится неотъемлемой частью школьной программы. Уже с начальных классов детей учат программировать роботов [32]. Обучение начинается с игровой формы, но позднее ребята начинают работать с реальными моделями и участвуют в соревнованиях по робототехнике.

Одним из инструментов, предназначенных для обучения кибернетике через программирование роботов, является TRIK Studio [3]. Это среда визуального программирования роботов с интерактивным режимом имитационного моделирования.

TRIK Studio предоставляет возможности для программирования различных конструкторов, включая Lego Mindstorms NXT и EV3, TRIK [24], а также разных моделей квадрокоптеров Геоскан Пионер [31].

Независимо от целевых платформ, которые сильно различаются архитектурно, программа в TRIK Studio представляется графически: в виде диаграммы из блоков (операторов) и стрелок, задающих порядок выполнения этих операторов. Получаемая из диаграммы программа в пригодном для исполнения на конкретной платформе виде может быть запущена на роботе одним нажатием кнопки «Выполнить».

Наглядность представления программы и простота интерфейса считаются ключевыми достоинствами TRIK Studio. Однако упомянутая функциональность загрузки программ на роботов не реализована для квадрокоптера Геоскан Пионер Мини [30].

# 1. Постановка задачи

Целью работы является реализация в TRIK Studio функциональности для загрузки программ на квадрокоптер Геоскан Пионер Мини. Для её выполнения были поставлены следующие задачи:

1. Провести обзор средств для загрузки программы на квадрокоптеры Пионер;
2. Собрать и запустить TRIK Studio;
3. Выбрать способ реализации необходимой функциональности;
4. Реализовать скрипт для загрузки программ с использованием библиотеки `pioneer_sdk` [9];
5. Протестировать полученный скрипт в TRIK Studio;
6. Оформить результат в качестве доступного для пользователя инструмента.

## 2. Обзор

Рассмотрим, как на данный момент происходит загрузка программы на квадрокоптер, какие инструменты можно использовать для автоматизация этого процесса, какие технологии были использованы для настройки программ и реализации инструмента загрузки.

### 2.1. TRIK Studio и Pioneer Station

После того, как пользователь составил программу в TRIK Studio и сгенерировал по ней код, его можно переносить на Пионер Мини. Это квадрокоптер начального уровня, предназначенный для обучения робототехнике и используемый в помещениях. Он идеально подходит детям, так как безопасен и прост для освоения.

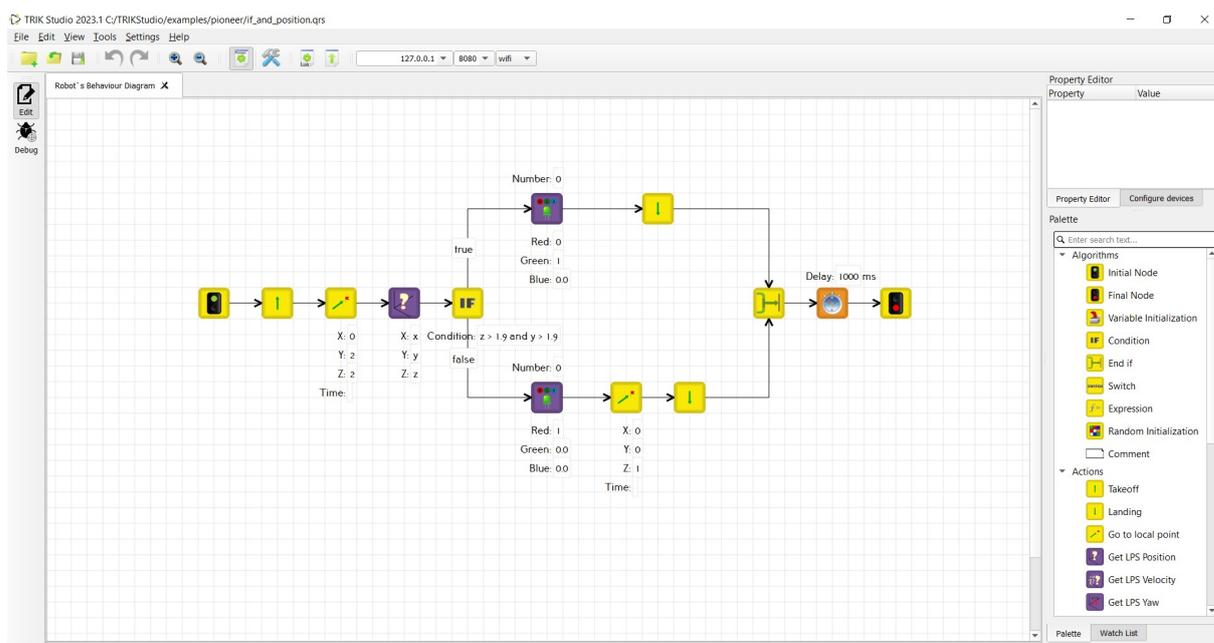


Рис. 1: Интерфейс приложения TRIK Studio

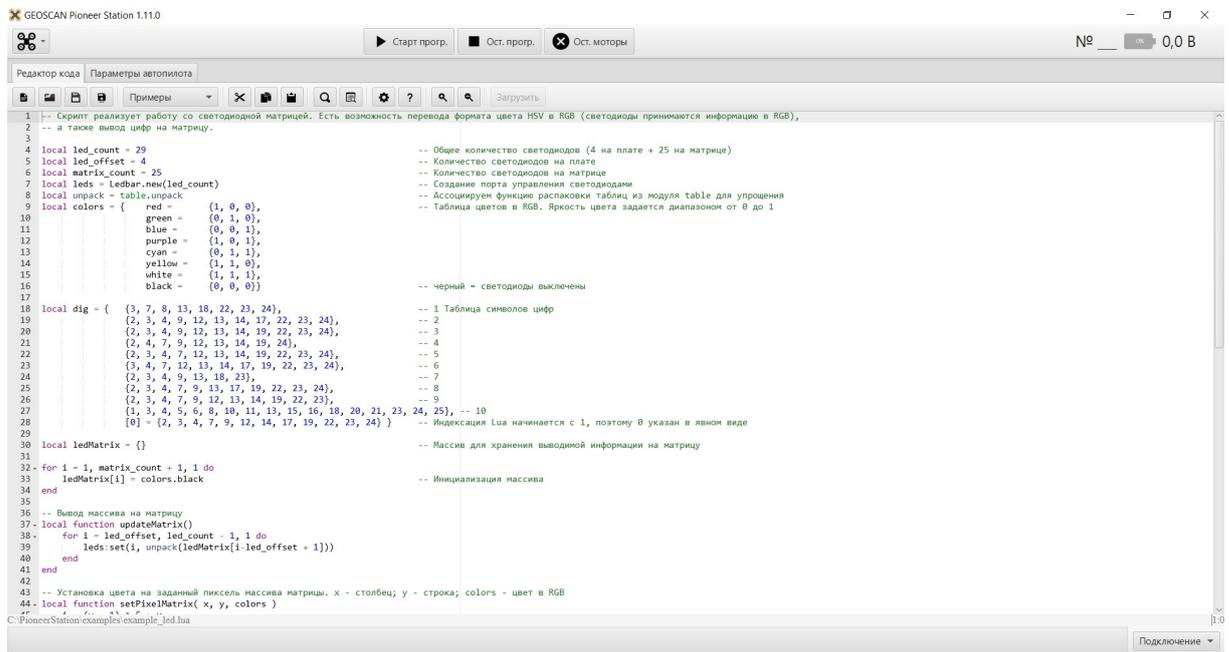
В данный момент пользователи испытывают трудности на этом этапе работы: в TRIK Studio есть кнопка загрузки, но нет отвечающей ей функциональности — пока что при нажатии на кнопку предлагается скачать еще одно приложение — Pioneer Station [17] — и производить загрузку через него.

## Errors

 [19:06:04]: **INFORMATION:** Please download uploader from [site](#)

Рис. 2: Сообщение о необходимости скачать Pioneer Station после попытки загрузить программу на квадрокоптер через TRIK Studio

Pioneer Station — это среда разработки, предназначенная для управления квадрокоптерами и другими беспилотными летательными аппаратами, производимыми компанией Геоскан. Pioneer Station прекрасно справляется с задачей загрузки, в этом приложении можно обновить прошивку или настроить параметры автопилота, взаимодействовать с различными устройствами (не только TRIK-овскими), используя разные протоколы связи, а также использовать это приложения при проведении научных исследований. Pioneer Station предоставляет обширный набор инструментов для разработки сложных проектов, однако пользователя TRIK Studio, который просто хочет перенести программу с компьютера на робот, лишняя функциональность будет только отвлекать от исходной цели запуска Pioneer Station.



```
1 -- Скрипт реализует работу со светодиодной матрицей. Есть возможность перевода формата цвета HSV в RGB (светодиоды принимают информацию в RGB),
2 -- а также вывод цифр на матрицу.
3
4 local led_count = 29 -- Общее количество светодиодов (4 на плате + 25 на матрице)
5 local led_offset = 4 -- Количество светодиодов на плате
6 local matrix_count = 25 -- Количество светодиодов на матрице
7 local leds = ledbar.new(led_count) -- Создание порта управления светодиодами
8 local unpack = table.unpack -- Ассоциируем функцию распаковки таблиц из модуля table для упрощения
9 local colors = { red = {1, 0, 0}, -- Таблица цветов в RGB. Яркость цвета задается диапазоном от 0 до 1
10 green = {0, 1, 0},
11 blue = {0, 0, 1},
12 purple = {1, 0, 1},
13 cyan = {0, 1, 1},
14 yellow = {1, 1, 0},
15 white = {1, 1, 1},
16 black = {0, 0, 0} -- черный = светодиоды выключены
17
18 local dig = { {3, 7, 8, 13, 18, 22, 23, 24}, -- 1 Таблица символов цифр
19 {2, 3, 4, 9, 12, 13, 14, 17, 22, 23, 24}, -- 2
20 {2, 3, 4, 9, 12, 13, 14, 19, 22, 23, 24}, -- 3
21 {2, 4, 7, 9, 12, 13, 14, 19, 24}, -- 4
22 {2, 3, 4, 7, 12, 13, 14, 19, 22, 23, 24}, -- 5
23 {3, 4, 7, 12, 13, 14, 17, 19, 22, 23, 24}, -- 6
24 {2, 3, 4, 9, 13, 18, 23}, -- 7
25 {2, 3, 4, 7, 9, 13, 17, 19, 22, 23, 24}, -- 8
26 {2, 3, 4, 7, 9, 12, 13, 14, 19, 22, 23}, -- 9
27 {1, 3, 4, 5, 6, 8, 10, 11, 13, 15, 16, 18, 20, 21, 23, 24, 25}, -- 10
28 [0] = {2, 3, 4, 7, 9, 12, 14, 17, 19, 22, 23, 24} } -- Индексация Lua начинается с 1, поэтому 0 указан в явном виде
29
30 local ledMatrix = {} -- Массив для хранения выводимой информации на матрицу
31
32 for i = 1, matrix_count + 1, 1 do -- Инициализация массива
33 ledMatrix[i] = colors.black
34 end
35
36 -- Вывод массива на матрицу
37 local function updateMatrix()
38 for i = led_offset, led_count - 1, 1 do
39 leds:set(i, unpack(ledMatrix[i-led_offset + 1]))
40 end
41 end
42
43 -- Установка цвета на заданный пиксель массива матрицы. x - столбец; y - строка; colors - цвет в RGB
44 local function setPixelMatrix(x, y, colors)
45 end
46
47 C:\PioneerStation\examples\example_led.lua
Подключение
```

Рис. 3: Интерфейс программы Pioneer Station

Итак, после того, как программа в TRIK Studio составлена, код сгенерирован, а Pioneer Station установлена, пользователю необходимо перенести код в это приложение любым удобным ему способом, например:

- сохранить файл с программой на компьютер, после этого открыть его в Pioneer Station;
- напрямую скопировать и вставить код из окна TRIK Studio в окно Pioneer Station.

После открытия файла в Pioneer Station, пользователь нажимает на кнопку «Загрузить», и программа переносится на Пионер Мини.

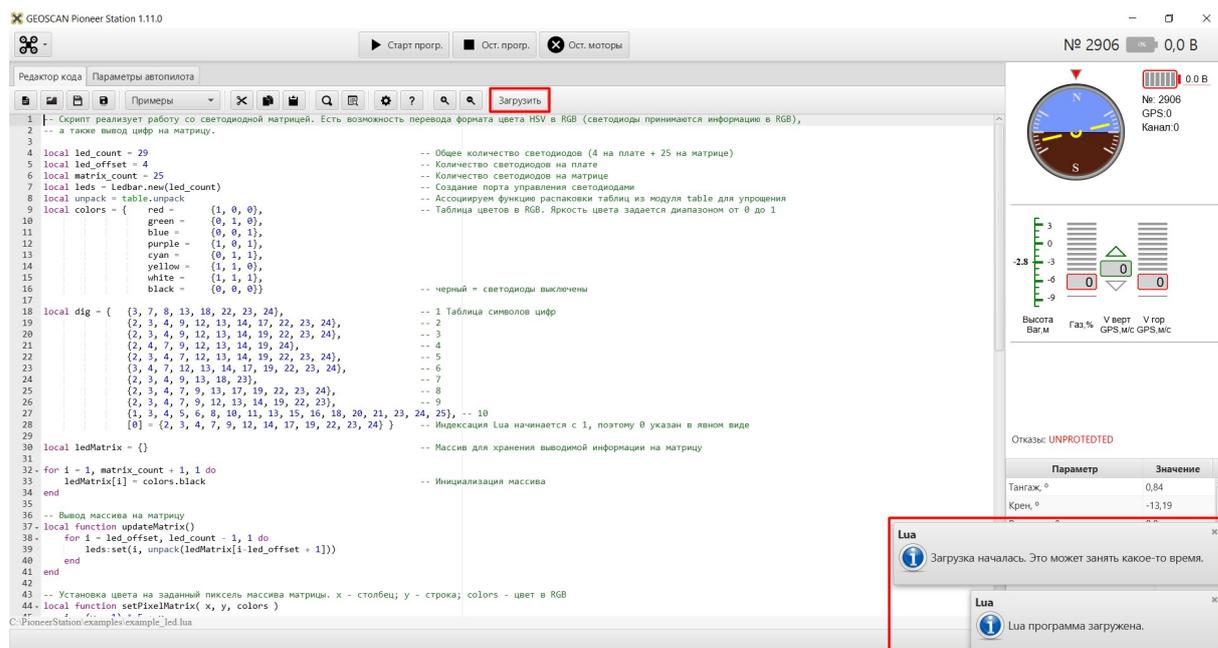


Рис. 4: Загрузка программы на Пионер Мини из Pioneer Station

Задача выполнена, однако для ее достижения требуется совершить много лишних действий. Также на протяжении всего процесса разработки у пользователя должны быть параллельно открыты две программы с примерно одинаковой логикой работы.

## 2.2. Pioneer SDK (Software Development Kit)

На данный момент для взаимодействия Пионера Мини с компьютером предусмотрена библиотека `pioneer_sdk` [9]. Этот набор инструментов предоставляется компанией Геоскан, и с его помощью можно создавать собственные программы для управления, мониторинга и интеграции с квадрокоптерами Геоскан Пионер. В том числе там есть API

для загрузки, запуска и останова программ, написанных на языке программирования Lua, что подходит для решения поставленной задачи.

В качестве альтернативы `pioneer_sdk` можно реализовать вручную и использовать для загрузки программ свой собственный протокол, однако при обсуждении задач с разработчиками TRIK Studio было принято решение, что брать новый код на сопровождение и писать к нему тесты — избыточно, так как уже существует библиотека, предоставляющая подходящую функциональность.

В результате при реализации скрипта была использована библиотека `pioneer_sdk`.

### 2.3. Используемые технологии

Для сборки и навигации в коде приложения TRIK Studio использовался Qt Creator [4]. Причина выбора этой интегрированной среды разработки вместо аналога KDevelop [12] — Qt Creator оптимизирован для работы с Qt [5], что соответствует нашим требованиям, так как TRIK Studio базируется на этом фреймворке. Также, TRIK Studio собирается с использованием QMake [23], а Qt Creator предоставляет интеграцию с этой утилитой, что упрощает настройку и сборку проекта. TRIK Studio является кросс-платформенным приложением, так что очередным плюсом использования Qt Creator-а является поддержка разработки и тестирования приложений на разных платформах.

Для разработки скрипта загрузки был выбран язык программирования Python [28], так как API для работы с квадрокоптером Пионер Мини на данный момент актуально только для Python-библиотеки `pioneer_sdk`.

Для написания скрипта использовалась среда разработки PyCharm [21], потому что она предоставляет богатый набор инструментов для разработки и отладки Python-приложений, в нее можно легко установить необходимые плагины и расширения, интегрироваться с инструментами анализа кода (`black` [2]), сборки (`tox` [22], `pytest` [11]) и Github-ом, к тому же, в документации `pioneer_sdk` указана именно эта среда

программирования.

Для управления пакетами использовался инструмент `pip` [27], так как в отличие от своего аналога `Poetry` [1], он уже установлен в стандартную библиотеку Python и не требует дополнительной установки и настройки инструментов. К тому же, функциональность `Poetry` избыточна для небольших проектов.

Для тестирования использовалась библиотека `pytest`, так как по сравнению с другим популярным модулем `unittest` [8], тесты в `pytest` пишутся более компактно, а проверка результатов более подробная.

Также для автоматического тестирования был использован инструмент `tox`, так как он:

- позволяет автоматически разворачивать и тестировать проекты (как локально, так и в CI);
- настраивается с помощью одного файла конфигурации;
- позволяет тестировать приложение на нескольких версиях Python.

## 3. Реализация

Перед началом разработки необходимо было приобрести базовые навыки работы с квадрокоптерами, локально собрать TRIK Studio и определиться со способом реализации инструмента.

### 3.1. Подготовка и настройка окружения

Для корректной работы всех систем Пионера Мини необходимо установить актуальную версию прошивки автопилота. Процедура обновления и настройки происходит с помощью программы Pioneer Station. Кроме этого, нужно обновить прошивку контроллера ESP-32: он отвечает за передачу видеопотока, а также за передачу команд управления с телефона на квадрокоптер. Для обновления прошивки ESP-32 необходимо воспользоваться одной из специальных утилит: ESPTool [14] или Njet [15]. Работа с утилитами осложнялась тем, что они всё еще в разработке: для корректной прошивки потребовалось время и правильные файлы. Также теперь представители компании Геоскан знают о проблемах с инструментами и собираются решить их в ближайшее время.

Для того, чтобы начать разработку, нужно было собрать приложение TRIK Studio локально. Так как в данной продуктовой задаче конечный пользователь использует операционную систему Windows, было решено использовать эту же платформу. Сборку пришлось проводить, основываясь только на командах из файлов CI репозитория [25], так как в проекте TRIK Studio этот процесс практически не задокументирован. По итогам практики я описала свой опыт сборки TRIK Studio из консоли, пулл-реквест был принят в официальный репозиторий приложения.

Для дальнейших действий со студией нужно было использовать Qt Creator. В процессе сборки в этой среде разработки потребовалось:

- Изучить инструмент QMake, который упрощает сборку Qt-проектов, предоставляя простой и понятный синтаксис и интеграцию с Qt-библиотекой, и используется для сборки TRIK Studio;

- Ознакомиться с динамическими библиотеками (.dll файлами), в том числе разобраться с проблемой DLL hell-a [29]: TRIK Studio находила и привязывала к себе неправильные файлы, называющиеся также, как нужные. Для решения этой проблемы был изучен порядок загрузки динамических библиотек [18], была проведена настройка параметров компьютера, выставлены правильные пути, в этом сильно помогло приложение Dependencies [6];
- Кроме этого, пришлось разобраться в методологии сборки крупных программных проектов в целом, и в коде TRIK Studio в частности. В процессе выполнения вышеописанных задач был приобретен некоторый объем знаний о языке программирования C++ и использовании классов из библиотеки Qt.

После выполнения этих шагов приложение TRIK Studio было собрано.

## 3.2. Выбор способа реализации

Вначале рассматривался вариант реализации функциональности через Python-скрипт, но тогда для его работы была бы необходима конкретная версия этого языка программирования. В TRIK Studio нет внутренней поддержки Python, соответственно, для использования инструмента пришлось бы брать версию, установленную на компьютер пользователя. И если она внезапно не подходит, нужно просить пользователя установить нужную, и не факт, что он сделает это правильно. К тому же, `pioneer_sdk` сопровождается отдельно, и добавлять лишнюю работу по сопровождению разработчикам TRIK Studio было бы нерационально.

TRIK Studio позиционируется как самостоятельное приложение, поэтому из обсуждений с разработчиками TRIK-а по опыту эксплуатации Геоскан Пионер было сформулировано следующее требование: реализовать скрипт именно как отдельный инструмент, упаковав его в исполняемый файл, который не будет зависеть от дополнительного программного обеспечения. Это также позволило не менять код TRIK Studio,

так как в нем уже есть функция, отвечающая за загрузку программы и принимающая необходимые параметры. Еще один плюс реализации загрузчика через исполняемый файл в том, что его можно как независимо вызывать из TRIK Studio, так и использовать отдельно, например, из консоли, если вдруг захотим загрузить готовую программу на подключенный дрон, или в других учебных проектах, связанных с квадрокоптерами Пионер Мини.

### 3.3. Реализация инструмента TRIK Studio Pioneer Tool

По итогам практики был написан Python-скрипт [26], который работает следующим образом:

1. Импортирует необходимые библиотеки;
2. Получает аргументы командной строки, которые ожидаются в следующем порядке: путь к файлу программы, IP-адрес дрона, порт, режим соединения (wifi или usb);
3. Далее идут проверки корректности данных: существование файла по заданному пути, правильный формат IP-адреса и порта, проверка на то, что режим соединения является одним из ожидаемых;
4. После этого скрипт создает экземпляр класса Pioneer из библиотеки pioneer\_sdk. Этот объект используется для взаимодействия с квадрокоптером;
5. Далее скрипт передает файл для загрузки на квадрокоптер;
6. Засыпает на две секунды, чтобы гарантированно успеть загрузить программу, а затем закрывает соединение с квадрокоптером;
7. Если в процессе выполнения кода возникает исключение, происходит запись сообщения об ошибке в лог, а далее пользователю поступает предложение скачать аналог загрузчика (Pioneer Station) с указанной веб-страницы [17].

Интерфейс TRIK Studio предоставляет два способа подключения квадрокоптера, и, соответственно, два способа загрузки программы: по usb-модему или по wifi. sdk же предоставляет возможность соединения с квадрокоптером тремя способами, которые отличаются типами взаимодействия: udrout, udrin и serial. Для соединения по wifi я использовала udrout. Этот тип соединения используется для отправки данных на удаленное устройство по UDP (User Datagram Protocol), что хорошо подходит для беспроводных соединений. Для модема был выбран тип serial, так как он используется для связи по последовательному порту (COM порту), что актуально для подключения через проводное соединение. udrin же используется для получения данных с квадрокоптера на компьютер через UDP-соединение, чего не требуется в моей задаче.

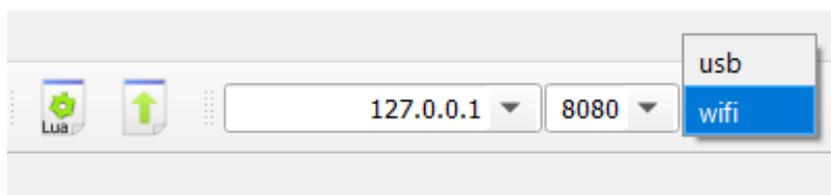


Рис. 5: Кнопка загрузки и выбор способа связи с дроном в TRIK Studio

Обработка ошибок в моей программе была оформлена через логирование, чтобы вписываться в логику TRIK Studio. Также для удобства пользователей к некоторым из них были добавлены более понятные описания.

При написании вышеописанного загрузчика использовалась библиотека `pioneer_sdk`, а именно — функция `lua_script_upload()` [16], отвечающая за загрузку Lua-скрипта на дрон. Однако, при работе с `pioneer_sdk` возникли следующие проблемы:

- Документация на официальном сайте компании Геоскан не обновлялась в соответствии с обновлениями библиотеки: некоторые примеры перестали быть актуальными и не соответствуют реальной функциональности библиотеки: их невозможно запустить, так как используемых классов и методов в актуальной версии библиотеки больше не существует;

- При запуске функции `lua_script_upload()` происходит ошибка, связанная с реализацией данной функции внутри библиотеки: на одном из этапов загрузки переданные данные обнуляются и не могут дойти до квадрокоптера;
- В библиотеке `pioneer_sdk` не настроена обработка ошибок: при возникновении непредвиденных ситуаций (например, не найден порт или дрон не подключен к компьютеру), программа бросает исключения из разных потоков. Это плохо, потому что пользователь видит всю трассировку стека вместо аккуратного и понятного сообщения о конкретной ошибке. Так как библиотека написана на языке программирования Python, возникаемые ошибки невозможно собрать в главном потоке без значительных изменений кода: нужно переделать весь процесс обработки ошибок.

Данный проект находится на стадии разработки, так что возникновение подобных ситуаций при использовании продукта вполне естественно. Обо всех вышеописанных сложностях было сообщено в компанию Геоскан. В планах сделать pull request-ы в репозиторий библиотеки и ожидать обновления компонентов.

Для локального использования данный скрипт нужно скачать с репозитория на Github [26] и единожды установить в папку, где находятся файлы TRIK Studio. После этого при нажатии на кнопку «Загрузить» в программе квадрокоптер всегда будет получать необходимые данные.

```

Errors
✘ [18:44:25]: ERROR: File "pioneer_sdk\piosdk.py", line 190, in _message_handler
✘ [18:44:26]: ERROR: File "pymavlink\mavutil.py", line 1103, in recv_msg
✘ [18:44:26]: ERROR: File "pymavlink\mavutil.py", line 1061, in recv
✘ [18:44:26]: ERROR: ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
✘ [18:44:26]: ERROR: 2023-10-07 18:44:25,600 - ERROR - [WinError 10054] An existing connection was forcibly closed by the remote host
✘ [18:44:26]: ERROR: 2023-10-07 18:44:25,600 - INFO - Please download uploader from site

```

Рис. 6: Пример результата вызова TRIK Studio Pioneer Tool из TRIK Studio

## 4. Дополнения к функциональности

В этом разделе рассмотрим подробнее оформление и тестирование созданного инструмента.

### 4.1. Тестирование

Так как Геоскан использует свой собственный протокол для передачи данных, а также они в любой момент могут изменить код, в реализуемый инструмент было решено добавить юнит тесты, проверяющие работоспособность программы.

В моем проекте инструмент tox [22] используется следующим образом: сначала файлы проекта проверяются с использованием инструментов flake8 [7], который анализирует код на наличие статических ошибок и проверяет его на соответствие стандарту PEP 8 [20], и black [2], который автоматически форматирует код в соответствии со стандартами оформления языка Python. В данном проекте тестирование происходило на минимально совместимой, дефолтной и последней (latest) версии. Под каждую конкретную версию Python создается виртуальная среда, в неё устанавливаются необходимые пакеты, а затем в каждой среде исполняется тестовый сценарий. Удобно, что все эти команды собраны в один инструмент, так что для исполнения тестов в Github Actions [10] достаточно написать одну строчку, а не прописывать отдельно каждое действие, вдобавок создавая отдельную матрицу версий Python для тестирования.

То есть, сейчас локальный запуск тестов доступен как с помощью библиотеки pytest, так и в более читаемом виде с помощью tox.

### 4.2. Оформление

Инструмент TRIK Studio Pioneer Tool был выложен на Github, в репозитории присутствует подробное README.md с описанием использования инструмента, настроен и проходит CI: проверяется соответствие кода стандартам языка, запускаются тесты, настроено обновление за-

висимостей проекта с использованием сервиса Dependabot [19]. Также при добавлении новой функциональности происходит автоматическая публикация новой версии инструмента в качестве артефакта сборки: актуальная версия всегда находится в папке `dist` ветки `dev`, где ведется разработка.

Автоматизация всех вышеперечисленных процессов значительно сокращает время настройки, тестирования, развертывания и сопровождения проекта. При желании данный проект можно собрать и запустить локально.

## Заключение

В результате работы над практикой был реализован загрузчик программ на квадрокоптер Геоскан Пионер. Инструмент оснащен документацией и тестами, выложен на Github [13], последний релиз доступен для пользователей во вкладке Releases. К тому же, pull request с функциональностью принят в репозиторий TRIK-а [26].

Были выполнены следующие задачи:

1. Проведен обзор доступных средств для загрузки программы на квадрокоптеры Пионер;
2. Опыт сборки и запуска TRIK Studio описан и добавлен в официальный репозиторий TRIK [25];
3. Выбран способ реализации и реализован скрипт для передачи программ на дрон с использованием библиотеки `pioneer_sdk`;
4. Полученный скрипт протестирован в TRIK Studio;
5. Результат оформлен в качестве отдельного инструмента.

В дальнейших планах:

1. Осуществлять поддержку и сопровождение продукта;
2. Добавить кроссплатформенность;
3. Уменьшить размер инструмента путем оптимизации кода.

Код инструмента доступен в GitHub-репозитории [13], разработка ведется под именем `kleo-53`.

## Список литературы

- [1] Acsany Philipp. Dependency Management With Python Poetry – Real Python. — URL: <https://realpython.com/dependency-management-python-poetry/> (online; accessed: 2023-10-07).
- [2] Black 23.9.1 documentation. — URL: <https://black.readthedocs.io/en/stable/> (online; accessed: 2023-10-07).
- [3] Bryksin Dmitry Mordvinov; Yurii Litvinov; Timofey. TRIK studio: Technical introduction. — URL: <https://ieeexplore.ieee.org/document/8071326> (online; accessed: 2023-10-07).
- [4] Company The Qt. Embedded Software Development Tools Cross Platform IDE | Qt Creator. — URL: <https://www.qt.io/product/development-tools> (online; accessed: 2023-10-07).
- [5] Company The Qt. Qt | Tools for Each Stage of Software Development Lifecycle. — URL: <https://www.qt.io/> (online; accessed: 2023-10-07).
- [6] Dependencies. — URL: <https://lucasg.github.io/Dependencies/> (online; accessed: 2023-10-07).
- [7] Flake8: Your Tool For Style Guide Enforcement. — URL: <https://flake8.pycqa.org/en/latest/index.html> (online; accessed: 2023-10-07).
- [8] Foundation Python Software. unittest — Unit testing framework — Python 3.12.0 documentation. — URL: <https://docs.python.org/3/library/unittest.html> (online; accessed: 2023-10-07).
- [9] Geoscan pioneer\_sdk. — URL: [https://github.com/geoscan/pioneer\\_sdk](https://github.com/geoscan/pioneer_sdk) (online; accessed: 2023-10-07).

- [10] GitHub Inc. Understanding GitHub Actions - GitHub Docs. — URL: <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions> (online; accessed: 2023-10-07).
- [11] Hillard Dane. Effective Python Testing With Pytest – Real Python. — URL: <https://realpython.com/pytest-python-testing/> (online; accessed: 2023-10-07).
- [12] KDE. KDevelop. A cross-platform IDE for C, C++, Python, QML/JavaScript and PHP. — URL: <https://kdevelop.org/> (online; accessed: 2023-10-07).
- [13] Karaseva Elizaveta. TRIK Studio Pioneer Tool. — URL: <https://github.com/kleo-53/trik-studio-pioneer-tool/releases> (online; accessed: 2023-10-07).
- [14] LTD Geoscan. Обновление прошивки ESP-32 — Документация Pioneer December update 2022 | Обновление с помощью ESPTool. — URL: <https://docs.geoscan.aero/ru/master/instructions/pioneer-mini/settings/esp32-update.html#esptool> (online; accessed: 2023-10-07).
- [15] LTD Geoscan. Обновление прошивки ESP-32 — Документация Pioneer December update 2022 | Обновление с помощью NJET. — URL: <https://docs.geoscan.aero/ru/master/instructions/pioneer-mini/settings/esp32-update.html#njet> (online; accessed: 2023-10-07).
- [16] LTD Geoscan. Описание методов pioneer\_sdk. — URL: [https://docs.geoscan.aero/ru/master/programming/python/pioneer-sdk-methods.html#lua\\_script\\_upload](https://docs.geoscan.aero/ru/master/programming/python/pioneer-sdk-methods.html#lua_script_upload) (online; accessed: 2023-10-07).
- [17] LTD Geoscan. Работа с Pioneer station. — URL: [https://docs.geoscan.aero/ru/master/programming/pioneer\\_station/pioneer\\_station\\_work.html](https://docs.geoscan.aero/ru/master/programming/pioneer_station/pioneer_station_work.html) (online; accessed: 2023-10-07).

- [18] Microsoft Documentstion. — URL: <https://learn.microsoft.com/en-us/windows/win32/dlls/dynamic-link-library-search-order#standard-search-order-for-unpackaged-apps> (online; accessed: 2023-10-06).
- [19] Mullans Alex. Keep all your packages up to date with Dependabot. — URL: <https://github.blog/2020-06-01-keep-all-your-packages-up-to-date-with-dependabot/> (online; accessed: 2023-10-07).
- [20] PEP 8 – Style Guide for Python Code | [peps.python.org](https://peps.python.org/pep-0008/). — URL: <https://peps.python.org/pep-0008/> (online; accessed: 2023-10-07).
- [21] PyCharm: all about the most popular Python IDE. — URL: <https://datascientest.com/en/pycharm-all-about-the-most-popular-python-ide> (online; accessed: 2023-10-07).
- [22] Samiullah Christopher. Python tox - Why You Should Use It and Tutorial. — URL: <https://christophergs.com/python/2020/04/12/python-tox-why-use-it-and-tutorial/> (online; accessed: 2023-10-07).
- [23] Smirnov Andrei. The Vital Guide to Qmake | Toptal. — URL: <https://www.toptal.com/qt/vital-guide-qmake> (online; accessed: 2023-10-07).
- [24] TRIK Kits. — URL: <https://trikset.com/en/prices> (online; accessed: 2023-10-14).
- [25] TRIK Studio. — URL: <https://github.com/trikset/trik-studio> (online; accessed: 2023-10-07).
- [26] TRIK Studio Pioneer Tool. — URL: <https://github.com/trikset/trik-studio-pioneer-tool> (online; accessed: 2023-10-07).

- [27] User Guide - pip documentation v23.2.1. — URL: [https://pip.pypa.io/en/stable/user\\_guide/](https://pip.pypa.io/en/stable/user_guide/) (online; accessed: 2023-10-07).
- [28] Welcome to Python.org. — URL: <https://www.python.org/> (online; accessed: 2023-10-07).
- [29] Wikipedia. DLL-Hell. — URL: [https://en.wikipedia.org/wiki/DLL\\_Hell](https://en.wikipedia.org/wiki/DLL_Hell) (online; accessed: 2023-10-07).
- [30] Геоскан Группа компаний. Геоскан Пионер Мини. — URL: <https://www.geoscan.ru/ru/products/pioneer/mini> (online; accessed: 2023-10-07).
- [31] Геоскан Пионер | ГК Геоскан. — URL: <https://www.geoscan.ru/ru/products/pioneer> (online; accessed: 2023-10-14).
- [32] Сушков Максим. Предмет для поколения «альфа»: как и зачем в школах учат робототехнике. — URL: <https://trends.rbc.ru/trends/education/644a1d0c9a7947de324cebdf> (online; accessed: 2023-10-07).