## Методические разработки учебных занятий для обучающихся технической направленности

# Программирование беспилотных летательных аппаратов DJI Tello

# 2024 год

Краткое описание:

Занятие посвящено получению первичных навыков программирования квадрокоптеров DJI Tello в среде Scratch.

Методическая разработка будет полезна педагогам дополнительного образования и учителям, использующим на занятиях квадрокоптеры DJI Tello. Как показал опыт, многие даже не догадываются о том, что эти квадрокоптеры можно программировать. А среди тех, кто знает о такой возможности – не все разобрались как это сделать. У меня никак не получалось сопряжение квадрокоптеров с редактором Scratch и лишь затратив довольно много времени на изучение англоязычных форумов посвящённых этой теме удалось понять все «тонкие места», своим опытом хочу поделиться со всеми интересующимися данной темой.

Тип занятия: изучение и первичное закрепление новых знаний.

Форма занятия: комбинированная.

Цели занятия:

Предметная: написать первую программу для квадрокоптера.

Методологическая: воспитание информационной культуры учащихся, развитие внимательности, логического мышления, развитие навыков коллективной работы и взаимопомощи.

Метапредметная: практическое применение программирования, развитие пространственного мышления

Методы обучения: объяснительно-иллюстративный, наглядный, частично-поисковый, исследовательский.

Ожидаемые результаты:

Учащиеся должны знать/понимать:

• назначение основных блоков программы в Scratch;

• комбинирование стандартных и специальных блоков;

• меры безопасности при работе с оборудованием.

Учащиеся должны уметь:

• подключить квадрокоптер к ноутбуку по Wi-Fi;

• запустить квадрокоптер согласно полётному заданию;

• выделять путь решения в зависимости от поставленной задачи.

Оборудование: квадрокоптер DJI Tello, ноутбук.

Используемые ЦОР: конспект занятия, презентация.

# Подготовительный этап

Перед началом занятия необходимо провести подготовительные мероприятия – установить и настроить программное обеспечение на ноутбуки обучающихся.

1. Скачать с официального сайта разработчика и установить бесплатный редактор Scratch 2.0 по ссылке <https://scratch.mit.edu/download/scratch2>. Необходимо учесть, что программирование квадрокоптеров возможно только в версии 2.0, последняя версия работать не будет.
2. Скачать и установить программную среду Node.js. Именно с её помощью программный код Scratch будет передаваться на исполнение квадрокоптеру. Следует учитывать, что здесь также нужно скачивать не самую последнюю версию, а 8.11.1

По ссылкам <https://nodejs.org/dist/v8.11.1/node-v8.11.1-x64.msi> для 64-разрядных ОС, или <https://nodejs.org/dist/v8.11.1/node-v8.11.1-x86.msi> для систем х86.

1. Скачать с сайта производителя квадрокоптеров набор файлов для взаимодействия между Scratch и Node. По ссылке  
   <https://dl-cdn.ryzerobotics.com/downloads/tello/Release.zip> . Эти файлы желательно разместить в каталоге C:\Tello или ином с простым коротким путём, так как этот путь придётся каждый раз набирать в консоли.

План занятия:

1. Организационный момент. Актуализация знаний (2 мин).
2. Теоретическая часть (8 мин).
3. Практическая часть (25 мин).
4. Вопросы учеников. Подведение итогов урока (3 мин).
5. Рефлексия (2 мин).

Ход занятия:

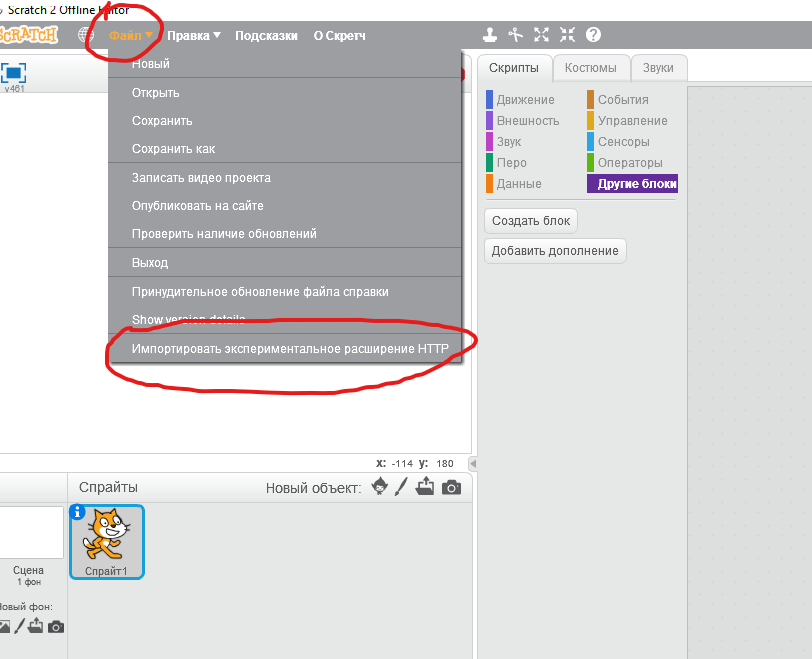
# **I. Организационный момент. Актуализация знаний.**

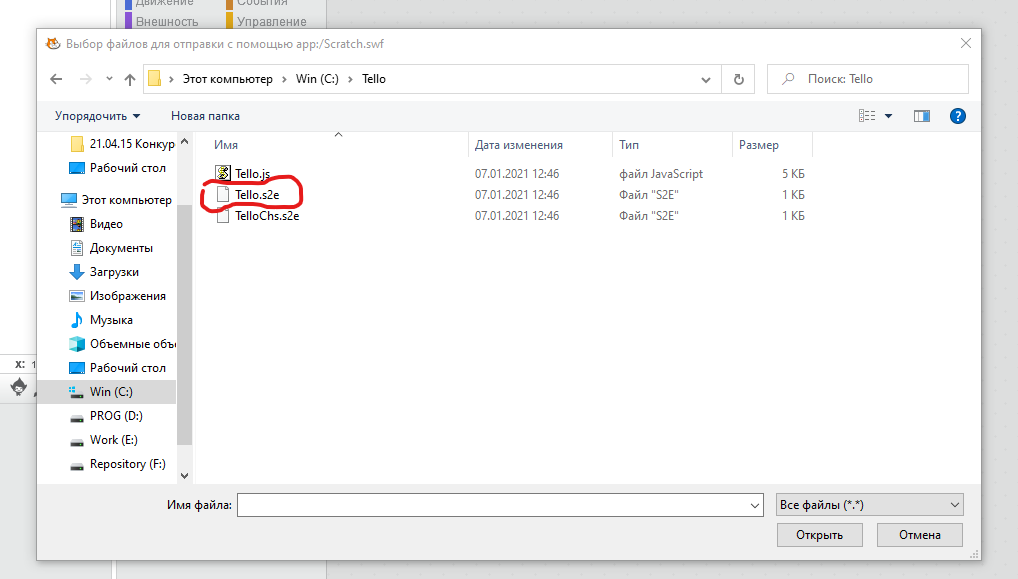
Здравствуйте, ребята. Сегодня мы с вами продолжим работу с квадрокоптерами DJI Tello. На предыдущих занятиях мы учились непосредственному пилотированию этих квадрокоптеров. Сегодня мы попробуем писать программу, которая будет управлять нашим беспилотным летательным аппаратом в соответствии с полётным заданием.

# **II. Теоретическая часть.**

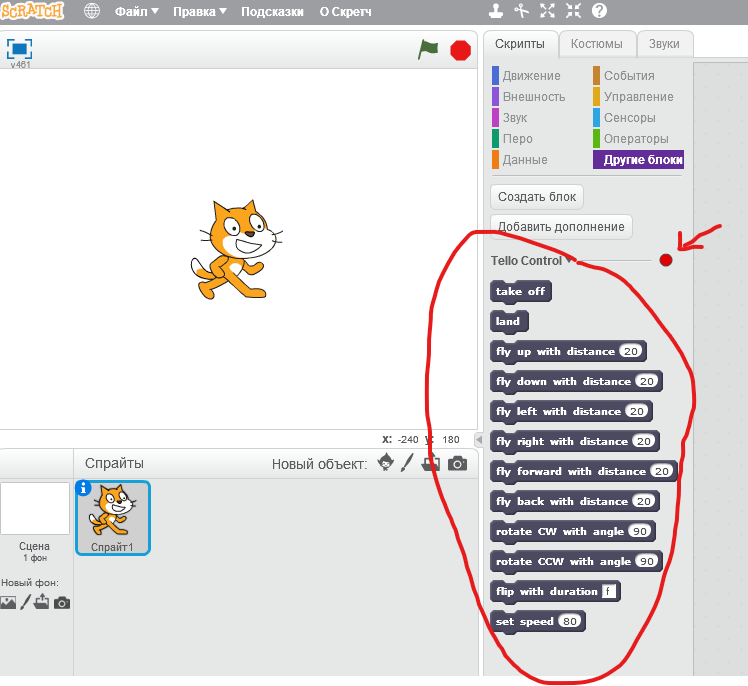
Запустите на своих ноутбуках редактор Scratch 2.0. Вы уже ранее работали в новой версии этого редактора, отличия в интерфейсе есть, но они незначительные.

Примечание для педагога: запустите презентацию на большом экране.

Выберите в разделе «скрипты» вкладку «другие блоки», как видите сейчас в ней пусто.Выберите меню «файл», изучите имеющиеся в нём вкладки. А теперь зажмите клавишу Shift на клавиатуре и ещё раз выберите меню «файл» - как видите появилась «секретная» вкладка «импортировать экспериментальные расширения HTTP» - нажмите на неё.

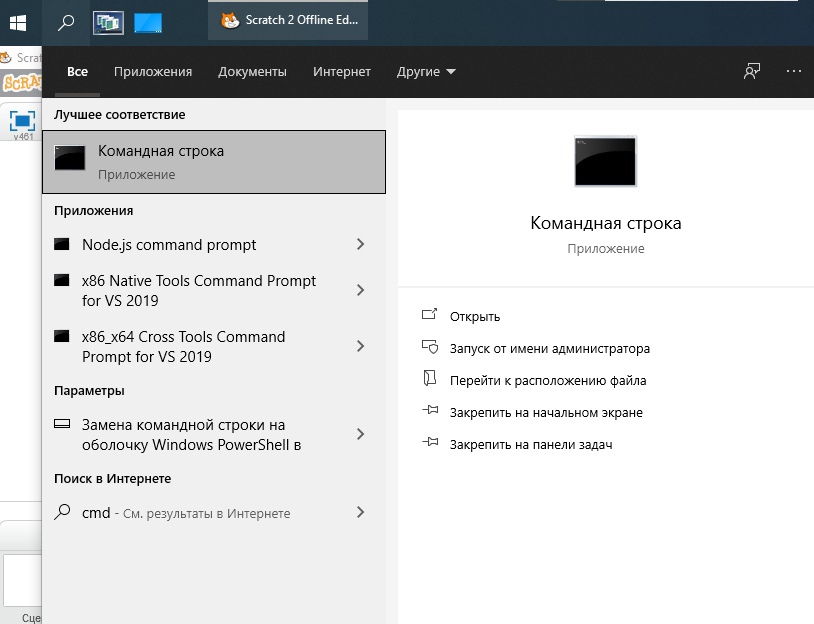
Перейдите в папку C:\Tello и запустите файл Tello.s2e.

Как видите во вкладке другие блоки появились блоки для управления квадрокоптером.



Все блоки подписаны на английском, но в них нетрудно разобраться.

Take off – взлёт. Land -посадка. Следующие блоки задают полёт или поворот в указанном направлении, примените свои знания, полученные на уроках английского языка, и переведите названия блоков.

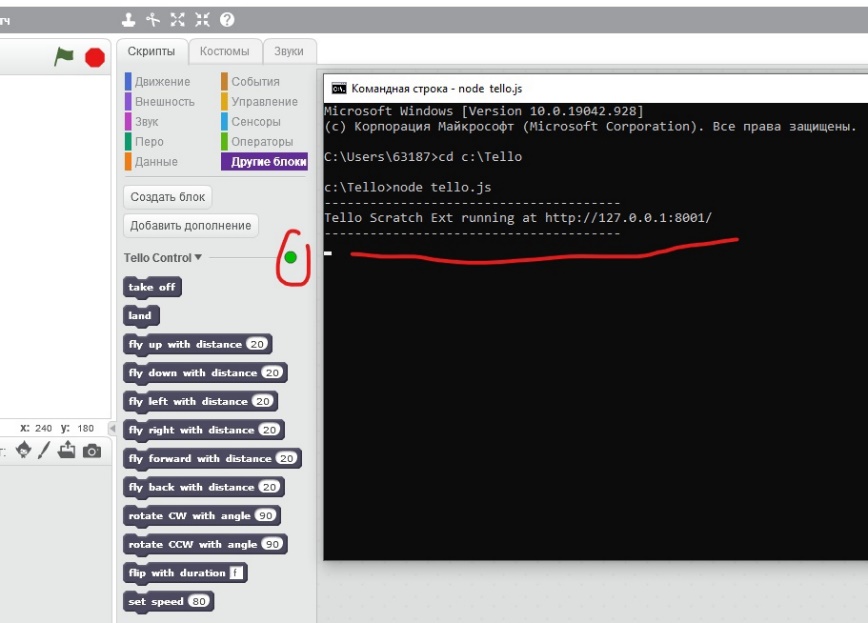
Обратите внимание на красный индикатор – он показывает, что пока система не готова к работе, нужно соединить редактор Scratch со средой Node которая будет пересылать команды нашему квадрокоптеру.

В строке поиска набираем латинскими буквами «cmd» - и вызываем командную строку. Далее, набираем команду cd c:\Tello.

Примечание для педагога: CD – это команда изменения директории (перехода к новой папке), а дальше адрес папки, в которую были скопированы файлы на подготовительном этапе.

После того, как появится строка c:\Tello набираем команду node tello.js

На экране должна появиться вот такая строка



И во вкладке дополнительные блоки индикатор должен поменять цвет с красного на зелёный – почти всё готово, программу уже можно писать.

# III. Практическая часть

Напишем первую простую программу для нашего квадрокоптера, а точнее сразу две – основную и аварийную.

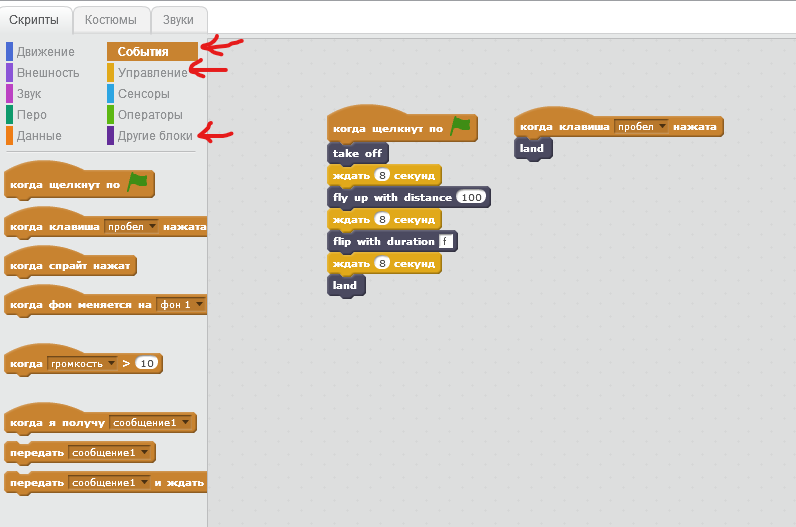
Основная программу будет такой:

1.Когда нажат зелёный флажок – взлететь (по этой команде квадрокоптер взлетает на высоту примерно 80 см.

2. Подняться на 100 см вверх (все измерения в этих блоках даны в сантиметрах).

3. Сделать флип (кувырок). Аргумент «f» у команды флип определяет, что кувырок будет сделан вперёд.

4. Приземлиться.

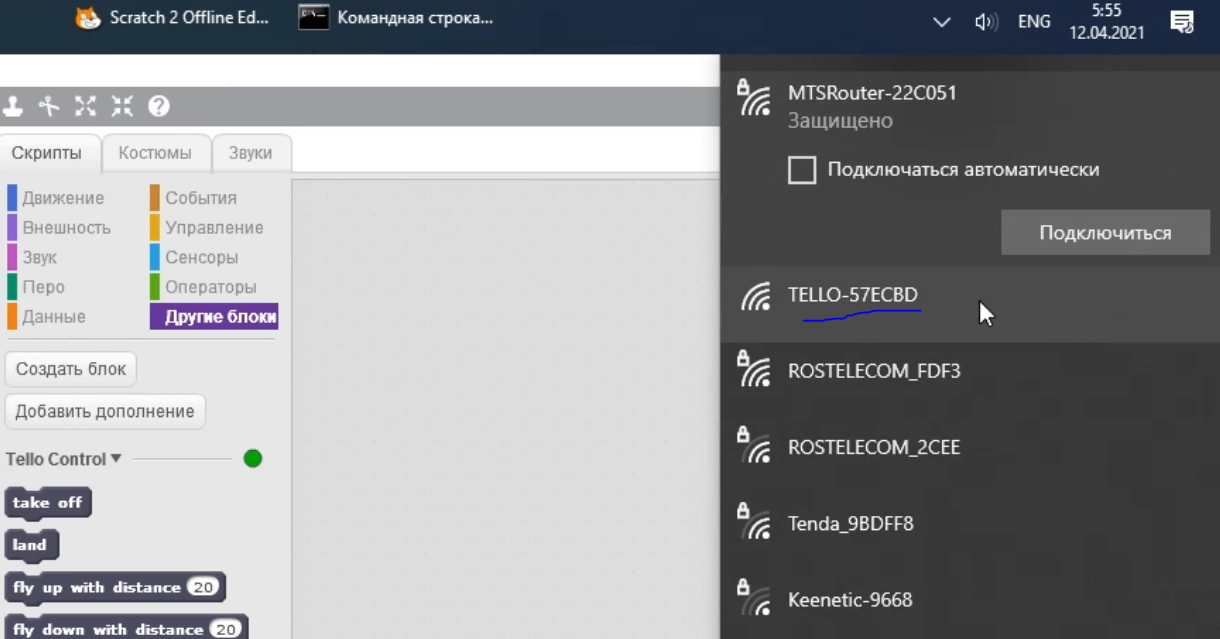
Обратите внимание – между всеми командами вставлена команда «ждать» с аргументом «8 секунд». Её приходится вставлять так как опытным путём мы установили, что иначе квадрокоптер не успевает спозиционироваться в новой точке и пропускает команды.

Обратите внимание, что рядом с основной имеется аварийная программа, которую желательно создавать всегда. Она очень простая – когда нажата клавиша «пробел» – приземлиться. Эта программа требуется на случай, если что-то пойдёт не так.

Теперь попробуем запустить наш коптер на выполнение программы.

Установим в него аккумулятор, включим и разместим на ровную поверхность. На занятиях по пилотированию вы уже узнали, что коптер очень плохо летает в слабоосвещённых помещениях, а также на тёмным или зеркальным напольным покрытием. Это связано с тем, квадрокоптер позиционируется по сигналу отражённом от пола и над такими типами напольной плитки летает плохо, его ведёт в сторону. При запуске по программе над такими покрытиями он не будет летать вовсе. Поэтому выбирает хорошо освещённые помещения со светлым матовым покрытием пола.

Подключимся к нашему квадрокоптеру по Wi-Fi



Как видите на поверхности квадрокоптера написаны две буквы или две цифры – это последние два символа его имени в сети Wi-Fi.

Примечание для педагога: лучше заранее подписать квадрокоптеры в соответствии с их именем Wi-Fi. На нижней части желательно сделать яркие цветные метки разного цвета на каждом или вовсе выкрасить нижнюю часть в яркий цвет – это удобно и при пилотировании – видно какой из квадрокоптеров летит куда-то не туда и можно быстро исправить ситуацию.

Выберите нужный и подключитесь. Как только вы увидите надпись «подключено» - запускайте программу на исполнение. Если разместить рядом с окном редактора Scratch окно консоли – можно в реальном времени наблюдать прохождение по сети исполняемых команд и заметить если какие-то команды пропущены.

Квадрокоптер взлетел, после паузы набрал высоту, выполнил флип и приземлился. Поэкспериментируйте с длительностью пауз в программе – возможно для ваших квадрокоптеров и ваших условий полёта паузы можно будет сократить или убрать вовсе. Если квадрокоптер пропускает какую-то команду – значит пауза слишком мала, он не успел провести позиционирование в пространстве.

Простейшую программу испытали – попробуем немного сложнее.

Задание: Квадрокоптер должен взлететь, подняться вверх на 100 см. Далее пролететь вперёд на 200 см, повернуться влево, ещё вперёд на 200 см, опять повернуться влево. И так последовательно прилететь в исходную точку.

Вам даётся 5 минут на написание программы.

Примечание для педагога. Желательно этот полёт провести вокруг какого-либо препятствия, например колонны в рекреации.

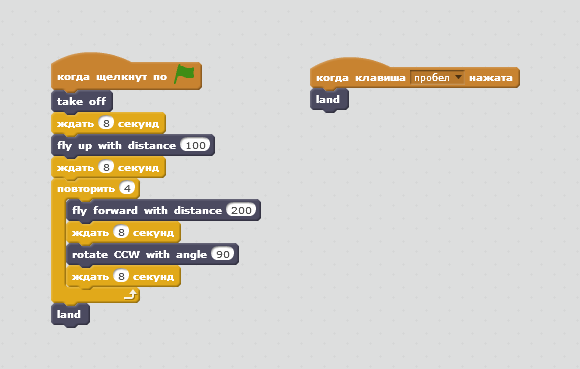
Обязательно проверяйте какую высоту дети указали в строке набор высоты, неверные значения могут привести к столкновению с потолком, что значительно опаснее для квадрокоптера нежели столкновение со стеной.

Проверим какие программы у вас получились.

Примечание для педагога. Возможны два пути решения этого задания – последовательное выполнение команд или цикл. Обратите внимание на тех детей, кто додумался до второго способа решения и отметьте их.

Обязательно создавайте аварийную программу. Квадрокоптер может просто зависнуть в воздухе в случае неправильного кода и висеть до разряда батарей. Его, конечно, можно аккуратно поймать снизу за корпус, но только в случае, если он завис на досягаемой высоте.

Вот один из вариантов решения задачи.



Примечание для педагога: Опытным путём мы установили, что команда лететь вперёд выполняется только на дистанцию менее 600 см.

На следующих занятиях мы попробуем написать программы для полётов по более сложным траекториям.

# IV. Вопросы учеников. Подведение итогов занятия.

Ответы на вопросы учащихся. Подведение итогов.

# V. Рефлексия.

Педагог ДО: Ребята, скажите, пожалуйста, своё мнение о нашем занятии, дополнив понравившиеся вам данные фразы своими мыслями. Оцените свою работу во время занятия.

сегодня я узнал...

было интересно...

было трудно...

я понял, что...

теперь я могу...

я научился...

я смог...

я попробую...

меня удивило...

урок дал мне для жизни...

мне захотелось...

Педагог ДО: спасибо! Вы замечательно выполнили все задания.