**Конспект практического занятия «Аэропилот»**

**для обучающихся 7 класса**

***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.

***Оборудование***: квадрокоптер DJI Tello, ноутбук.

***Используемые ЦОР***: конспект занятия, презентация.

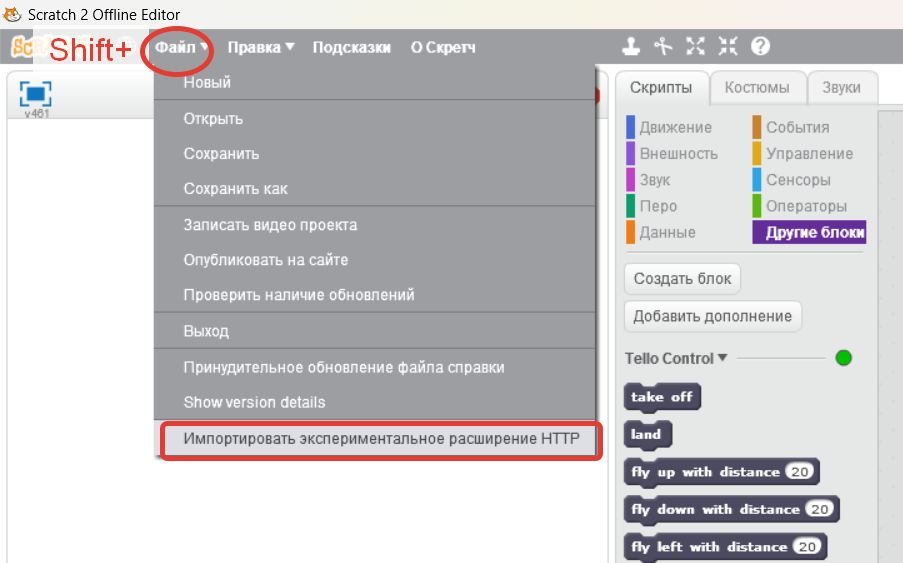
***Ход занятия:***

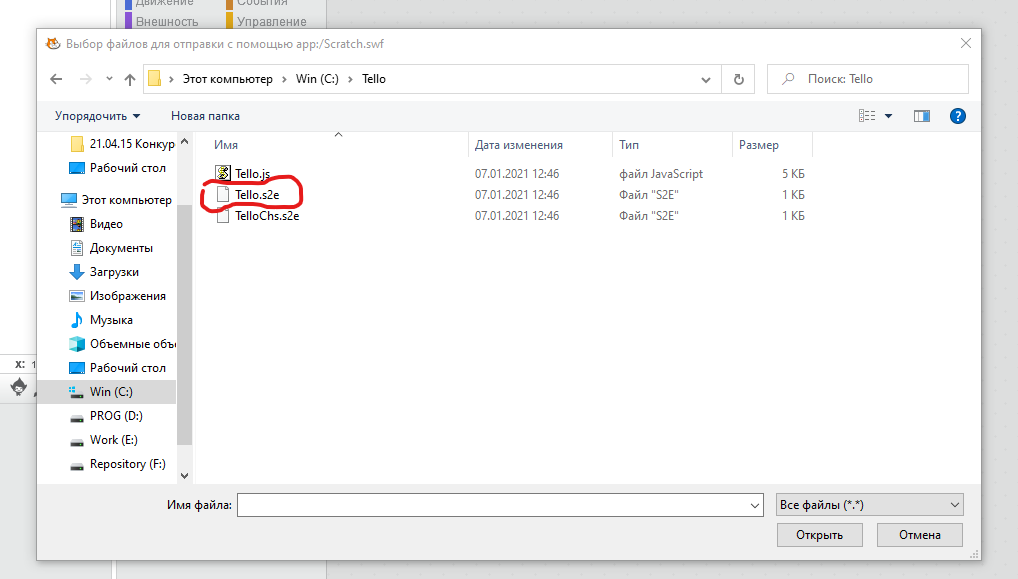
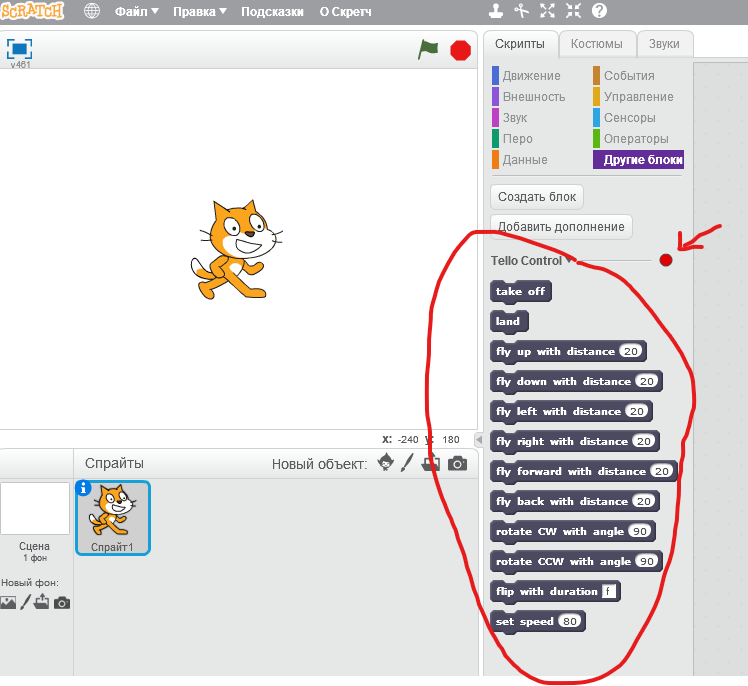
*I. Организационный момент. Актуализация знаний.*

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.

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.

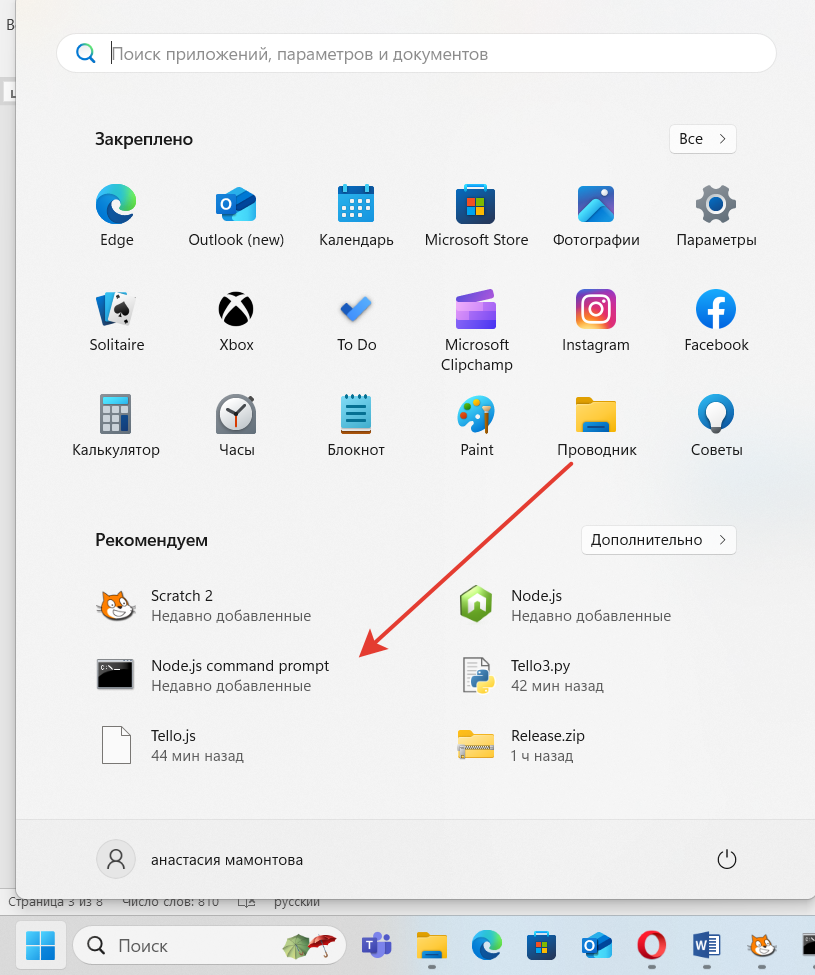
 Перейдите в папку C:\Tello и запустите файл Tello.s2e.

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

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

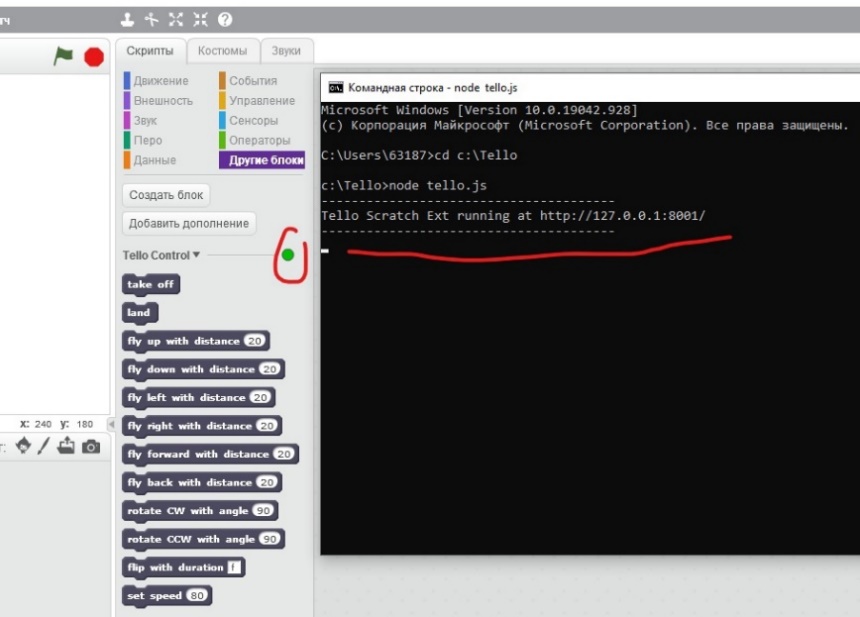
Take off – взлёт. Land -посадка. Следующие блоки задают полёт или поворот в указанном направлении, примените свои знания, полученные на уроках английского языка, и переведите названия блоков.

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.

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.

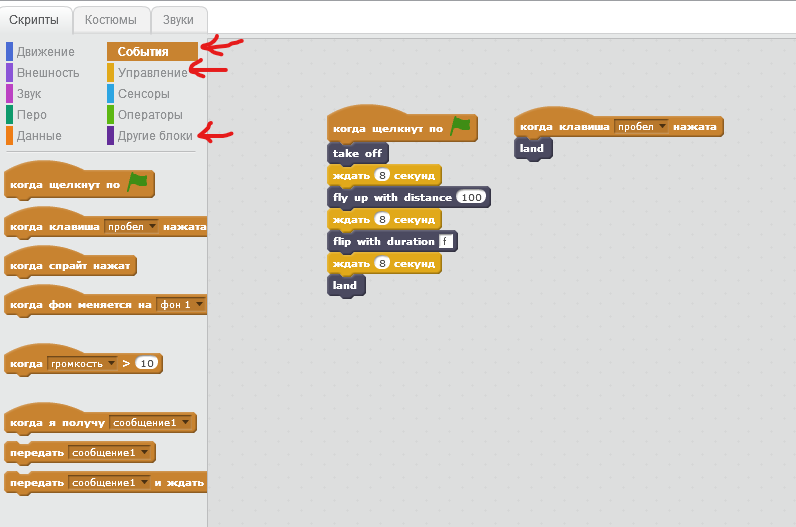
Основная программу будет такой:

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.

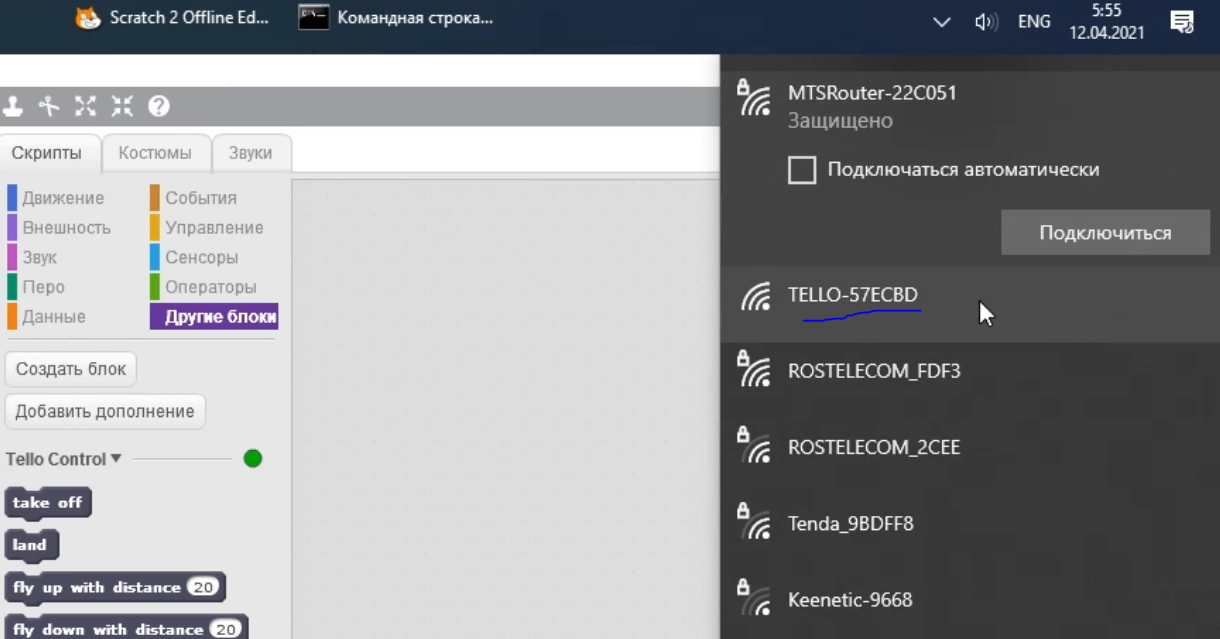
Обратите внимание – между всеми командами вставлена команда «ждать» с аргументом «8 секунд». Её приходится вставлять так как опытным путём мы установили, что иначе квадрокоптер не успевает спозиционироваться в новой точке и пропускает команды.

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.

Вспомним простые правила:

1. Установим в него аккумулятор;
2. Включим и разместим на ровную поверхность;
3. 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.

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.

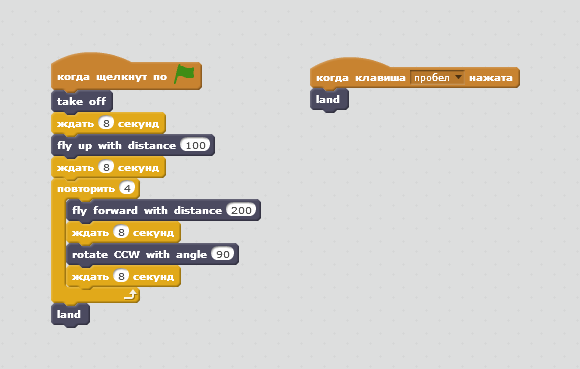
Простейшую программу испытали – попробуем немного сложнее.

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.

Проверим какие программы у вас получились.

Вот один из вариантов решения задачи.



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.**

Ребята, 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...

***Спасибо! Вы замечательно выполнили все задания.***