

УДК 519.687.7

<https://doi.org/10.36906/AP-2020/39>

УПРАВЛЕНИЕ КВАДРОКОПТЕРОМ С ПОМОЩЬЮ ВЕБ-САЙТА

Долгушин Д. А.

*Нижневартровский государственный университет
г. Нижневартовск, Россия*

Слива М. В.

*канд. пед. наук
Нижневартровский государственный университет
г. Нижневартовск, Россия*

Аннотация. В данной статье рассматривается управление квадрокоптером с помощью сайта, используя сервер, API дрона и голосовое управление. Рассмотрены некоторые библиотеки и код программы.

Ключевые слова: Tello, веб-сайт, квадрокоптер, SpeechRecognition, IoT.

«Интернет вещей» — следующий уровень развития устройств, которые могут объединяться в сеть через интернет или с помощью беспроводных технологий. Они обмениваются данными в режиме реального времени как напрямую, так и через удаленные онлайн-серверы (<https://clck.ru/T7w7n>).

В данной статье рассматривается создание управления квадрокоптером посредством готового API, передача данных между дроном и веб-сайтом, используя концепцию интернета вещей и голосовое управление с помощью встроенного в браузер Web Speech API.

В качестве беспилотного летательного аппарата будет использоваться Ryze Tello (рис. 1) — дрон от китайской частной компании DJI.



Рис. 1. Квадрокоптер

Квадрокоптер представляет собой точку доступа Wi-Fi. Для того, чтобы отправлять и принимать какие-либо данные с дрона, необходимо к нему подключиться. Для подключения существует множество различных приложений, в том числе и официальные от производителя.

У данного дрона уже есть готовое API для управления (рис. 2), поэтому остается только написать код, который бы передавал эти команды дрону.

Tello Commands

Control Commands		
Command	Description	Possible Response
Command	Enter SDK mode.	ok / error
takeoff	Auto takeoff.	
land	Auto landing.	
streamon	Enable video stream.	
streamoff	Disable video stream.	
emergency	Stop motors immediately.	
up x	Ascend to "x" cm. x = 20-500	
down x	down "x" Descend to "x" cm. x = 20-500	
left x	Fly left for "x" cm. "x" = 20-500	
right x	Fly right for "x" cm. "x" = 20-500	
forward x	Fly forward for "x" cm. "x" = 20-500	
back x	Fly backward for "x" cm. "x" = 20-500	

Рис. 2. Tello API

В первую очередь нужно создать сервер. Он будет реализован с помощью Node.js. Node.js — программная платформа, которая позволяет использовать JavaScript в качестве языка общего назначения. Добавляет возможность языку JavaScript взаимодействовать с устройствами ввода-вывода, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода [1].

Также понадобится фреймворк Express — используется в Node.js и предоставляет обширный набор функций.

В проект понадобится установить пару пакетов с помощью пакетного менеджера npm:

- express;
- socket.io — для обеспечения двусторонней связи на основе событий в режиме реального времени.

Для начала нужно получать с дрона данные, для этого нужно написать код для сервера в файле fly.js:

```

const dgram = require('dgram')
const app = require('express')()
const http = require('http').Server(app)
const io = require('socket.io')(http)

const PORT = 8889
const HOST = '192.168.10.1'

const drone = dgram.createSocket('udp4')
drone.bind(PORT)

const droneState = dgram.createSocket('udp4')
droneState.bind(8890)

drone.on('message', (message) => {
  console.log(`Drone: ${message}`)
})

```

```
droneState.on('message', (state) => {  
  console.log(`Drone: ${state}`)  
})  
  
http.listen(6767, () => {  
  console.log('Socket io server up and running')  
})
```

Теперь, после подключения к дрону через wi-fi, и, после запуска файла командой `node fly.js`, в консоль от дрона будет возвращаться ответ дрона на посылаемые ему команды (рис. 3).

```
running command: command  
👤 : ok  
running command: battery?  
👤 : 72  
  
running command: takeoff  
👤 : ok
```

Рис. 3. Ответ дрона

В `droneState` будет возвращаться состояние дрона: высота, крен, положение в пространстве, уровень заряда батареи и т.д. (рис. 4).

```
👤 : pitch:-24;roll:-19;yaw:122;vqx:0;vgy:0;vgz:0;templ:64;temph:66;tof:10;h:0;bat:63  
;baro:142.61;time:0;agx:-68.00;agy:389.00;agz:-953.00;
```

Рис. 4. Состояние дрона

Теперь можно попробовать отправить команды дрону, для этого в файл `fly.js` нужно дописать код:

```
const handleError = (err) => {  
  if (err) {  
    console.log('ERROR:', err)  
  }  
}  
  
io.on('connection', (socket) => {  
  socket.on('command', (command) => {  
    console.log('Command sent from browser:', command)  
    drone.send(command, 0, command.length, PORT, HOST, handleError)  
  })  
})
```

Когда произойдет событие "command", сработает функция обратного вызова и отправит дрону команду, ее длину, на указанный адрес. Также нужен обработчик ошибок `handleError`.

На стороне фронтенда нужно посылать команды на сервер, которые потом сервер отправит дрону, в зависимости от того, какая была нажата кнопка на клавиатуре. Для начала нужно добавить команды в отдельный файл `commands.js`:

```
const commands = {
  32: 'takeoff',
  27: 'land',
  87: 'forward 20',
  83: 'back 20',
}
module.exports = commands
```

Коды кнопок: 32 — пробел (взлет), 27 — esc (посадка), 87 — w (вперед), 83 — s (назад).

В новом файле `Commands.js` будет такой код:

```
import commands from '../commands/commands'
document.addEventListener('keydown', (event) => {
  for (let key in commands) {
    if (event.keyCode === +key) {
      console.log('Sending command', commands[key])
      socket.emit('command', commands[key])
    }
  }
})
```

Теперь, в зависимости от того, какая кнопка на клавиатуре нажата, выполнится соответствующая команда.

Также есть возможность голосового управления. Можно интегрировать на сайт готового голосового помощника, например, Alan AI. Но в данном случае хватит воспользоваться встроенным в браузер Web Speech API. Web Speech API позволяет взаимодействовать с голосовыми интерфейсами в ваших веб приложениях. Web Speech API состоит из двух частей: `SpeechSynthesis` (Текст-в-Речь) и `SpeechRecognition` (Асинхронное распознавание речи) (<https://clck.ru/T7vu7>).

Для голосового управления также понадобятся команды. Нужно создать файл `voiceCommands.js` и добавить команды:

```
const voiceCommands = {
  старт: 'takeoff',
  посадка: 'land',
}
module.exports = voiceCommands
```

В файл `Commands.js` нужно добавить голосовое управление:

```
const startSpeak = () => {
  const SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition
  var recognizer = new SpeechRecognition()
  recognizer.interimResults = true
  recognizer.lang = 'ru-RU'
  recognizer.onresult = function (event) {
    var result = event.results[event.resultIndex]
    if (result.isFinal) {
```

```
console.log(result[0].transcript)
console.log(voiceCommands[result[0].transcript])
socket.emit('command', voiceCommands[result[0].transcript])
}
}
recognizer.start()
}
```

Функцию `startSpeak` надо связать с кнопкой, по нажатию на которую браузер запросит доступ к микрофону. После предоставления доступа при голосовой команде «взлет» дрон взлетит.

Подводя итоги можно сказать, что дроны также используются в концепции интернета вещей, а управлять ими стало возможным через обычный браузер с помощью кнопок на сайте, на клавиатуре и голосом.

Литература

1. Слива М. В. Использование Node.js в качестве платформы для умного помещения // Традиции и инновации в образовательном пространстве России: материалы VII Всероссийской научно-практической конференции (г. Нижневартовск, 21 апреля 2018 года) / отв. ред. А.А. Никифорова. Нижневартовск: Изд-во Нижневарт. гос. ун-та, 2018. С. 49-52.

©Долгушин Д. А., Слива М. В., 2020