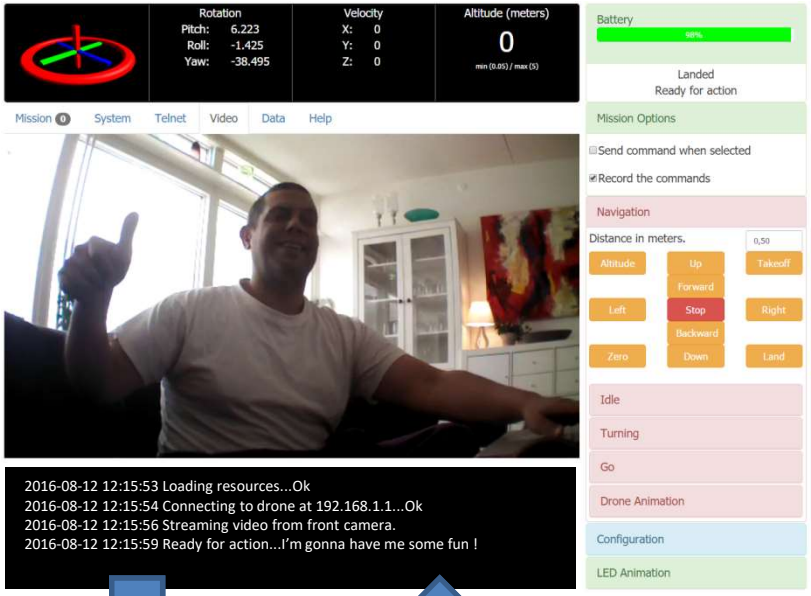
PROGRAMMING THE AR PARROT V2 DRONE USING NODEJS & JAVASCRIPT



WEBSOCKET EMIT COMMANDS & CONFIGURATION

WEBSOCKET EMIT DATA & STATUS

TELNET FTP

VIDEO SIGNAL DATA & STATUS

COMMANDS & CONFIGURATION

mydroneserver

DRONE CLIENT PROXY

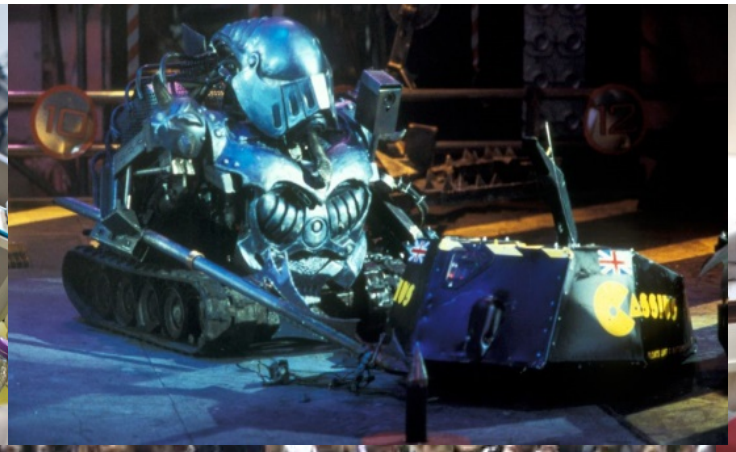
ar-drone

node

TechPlay

A social creative platform, where we explore technologies and make dreams come true


LESS OF THIS




WINNER


MORE OF THIS

WHERE DRONES ARE USED TODAY

- The military, firefighters and police
- Nature- and human disasters
- Doctors Without Borders
- Agriculture
- The insurance industry
- Photographers and media professionals
- Transport industry (control of vehicles, aircraft and ships)
- Construction industry
- Oil and natural resource industries
- The private toy & gadget market
- Postal and packaging industry
- Nature parks fighting poaching, animal control (drone guards / shepherds)
- Security and surveillance industry
- Property brokers
- Research and environmental monitoring
- Sports events
- Media and entertainment industry
- DIY: Do It Yourself products

WHERE ELSE COULD DRONES BENEFIT US?



THE HUMAN CARRIER DRONE IS HERE!

TOP DRONE PRODUCERS

CAMERA DRONES

DJI PHANTOM 4

DJI PHANTOM 3 SERIES

DJI INSPIRE 1 WITH 4K

YUNEEC Q500 4K

3DR SOLO

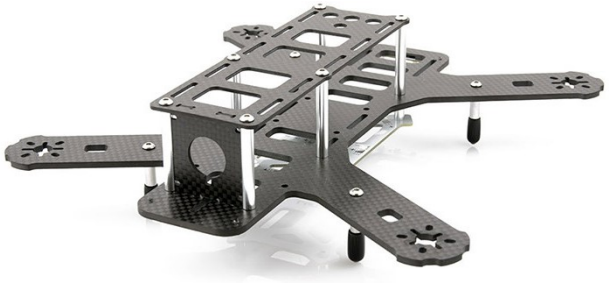
PARROT BEBOP

RACER DRONES

TBS VENDETTA

IMMERSIONRC VORTEX 285

IMMERSIONRC VORTEX 250 PRO



LUMENIER QAV250

EACHINE RACER 250

PLAY DRONES

SYMA X5C

HUBSAN X4

BLADE NANO QX

OUR DRONE FOR TODAY!

LATRAX ALIAS

PARROT AR DRONE 2.0

PROTO X

FREEFLIGHT - SMARTPHONE APP

AR PARROT V2 DRONE SPECS

1GHz 32 bit ARM Cortex A8 processor with
800MHz video DSP TMS320DMC64x
1GB DDR2 RAM at 200 MHz

Linux 2.6.32 (BusyBox)
Wi-Fi
USB 2.0 high speed for extensions

3 axis gyroscope 2000°/second precision
3 axis accelerometer +-50mg precision
3 axis magnetometer 6° precision
Pressure sensor +/- 10 Pa precision
Ultrasound sensors for ground altitude measurement
60 FPS vertical QVGA camera for ground speed measurement

AR PARROT DRONE V2

SYSTEM PORTS

Default IP: 192.168.1.1

Linux
BusyBox 2.6.32

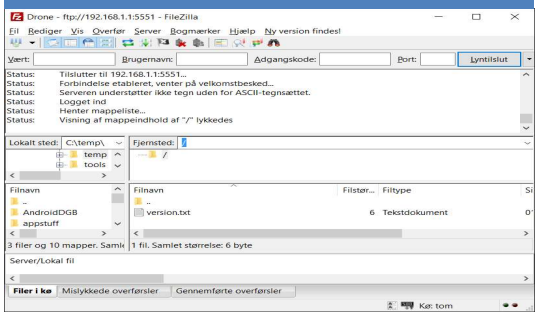

DATA USB


GPS USB

WI-FI HUB

Telnet port:23	Ftp port:5551	Navdata UDP port:5554	Video Stream port:5555	Control UDP port:5556	*Raw Capture port:5557*	*TCP Console port:5558*	*TCP Control port:5559*

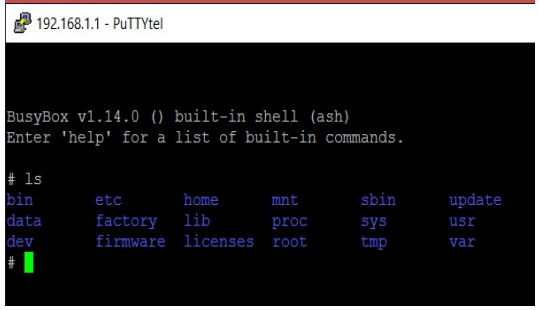
Ftp / FileZilla



AT* Commands

- **AT*REF**- for takeoff, landing, reset and emergency stop
- **AT*PCMD**- for motion, roll, pitch, gaz and yaw
- **AT*FTRIM**- sets the reference for the horizontal plane
- **AT*CONFIG**- configures the drone
- **AT*CONFIG_IDS**- used as identifiers for AT*CONFIG commands
- **AT*LED**- sets a LED animation on drone
- **AT*ANIM**- sets a flight animation gesture on the drone
- **AT*COMWDG**- resets the communication watchdog (debugging)
- And more…

Telnet / puTTYtel



cat /data/config.ini
cat /proc/cpuinfo
cat /proc/meminfo

Data messages (telemetry)


```
{
 "header": 1432778632,
 "droneState": {
 "flying": 0,
 "videoEnabled": 0,
 "visionEnabled": 0,
 "cameraReady": 1,
 "travellingEnabled": 0,
 "usbReady": 0,
 "motorProblem": 0,
 "communicationLost": 0,
 "softwareFault": 0,
 "lowBattery": 0,
  .....
```

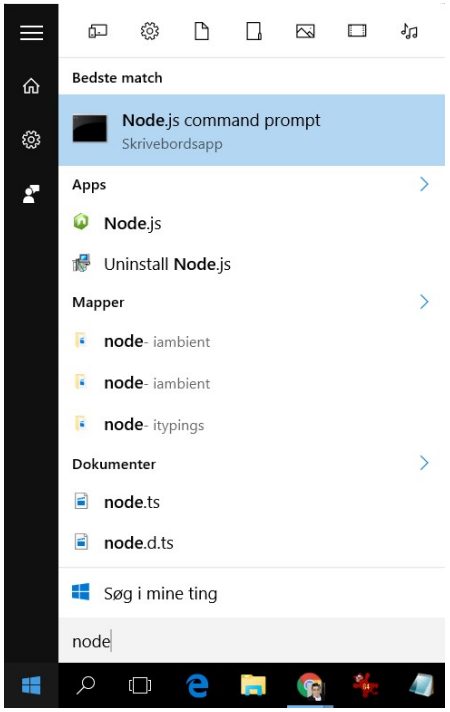
and lots of subscribable data sections

Video via Java, Javascript, FFmpeg...

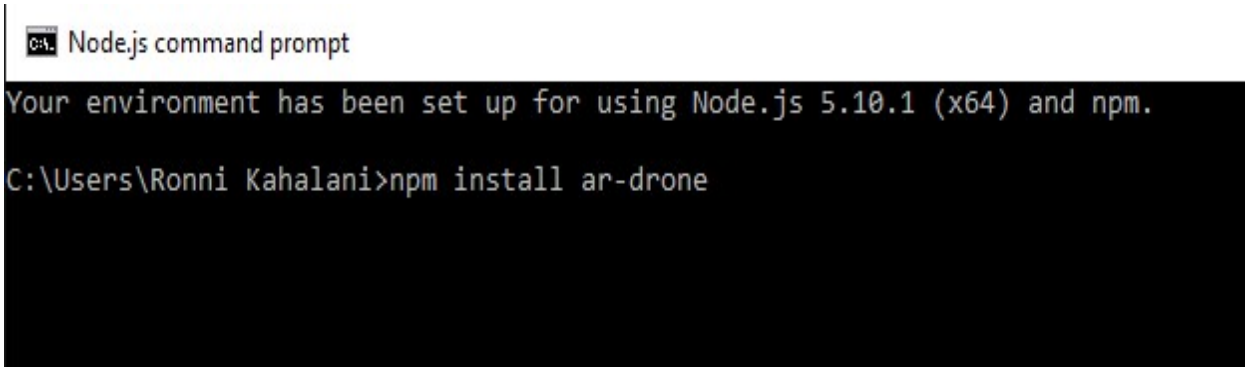
NODEJS AND JAVASCRIPT API

NodeJS is a popular plugin-based JavaScript server platform, which runs locally. Many developers know or have heard of it and it's a great platform for experiments. Download NodeJS, if it's not installed.

When NodeJS is installed, we need to install the ar-drone NodeJS plugin:



- Open a NodeJS console and change to your home folder
  - e.g. "C:\Users\<your username>", on Windows 10.
- Run: **npm install ar-drone**



Later on, you can try the more mission oriented control API

- ardrone-atonomy

NODEJS AND JAVASCRIPT API

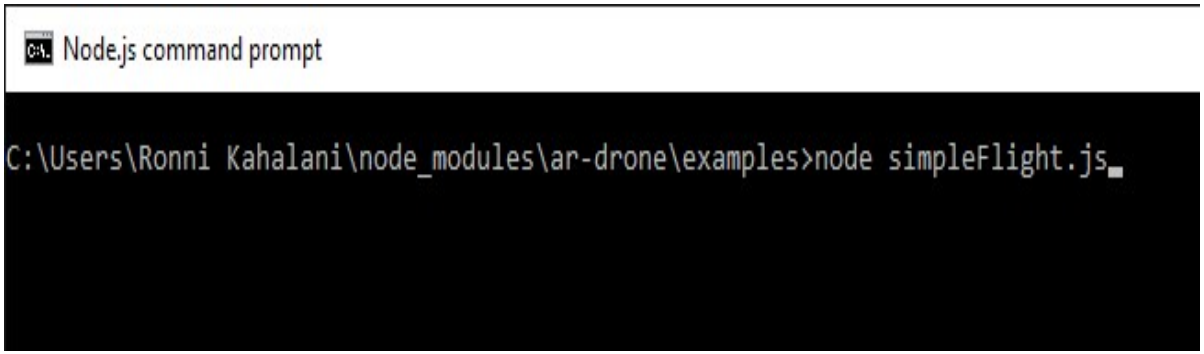
BEFORE RUNNING ANY DRONE CODE

- Make sure you have enough space for playing with the drone
- Create a NodeJS project, in your favorite IDE or a JavaScript file that runs in the context of NodeJS
- Prepare to leave the Internet, when you connect to the drone WiFi
- Connect the computer to the drone Wi-Fi
- Try to telnet (port:23) or ftp (port:5551) on the default host IP 192.168.1.1 (anonymously)

RUNNING THE AR-DRONE EXAMPLES

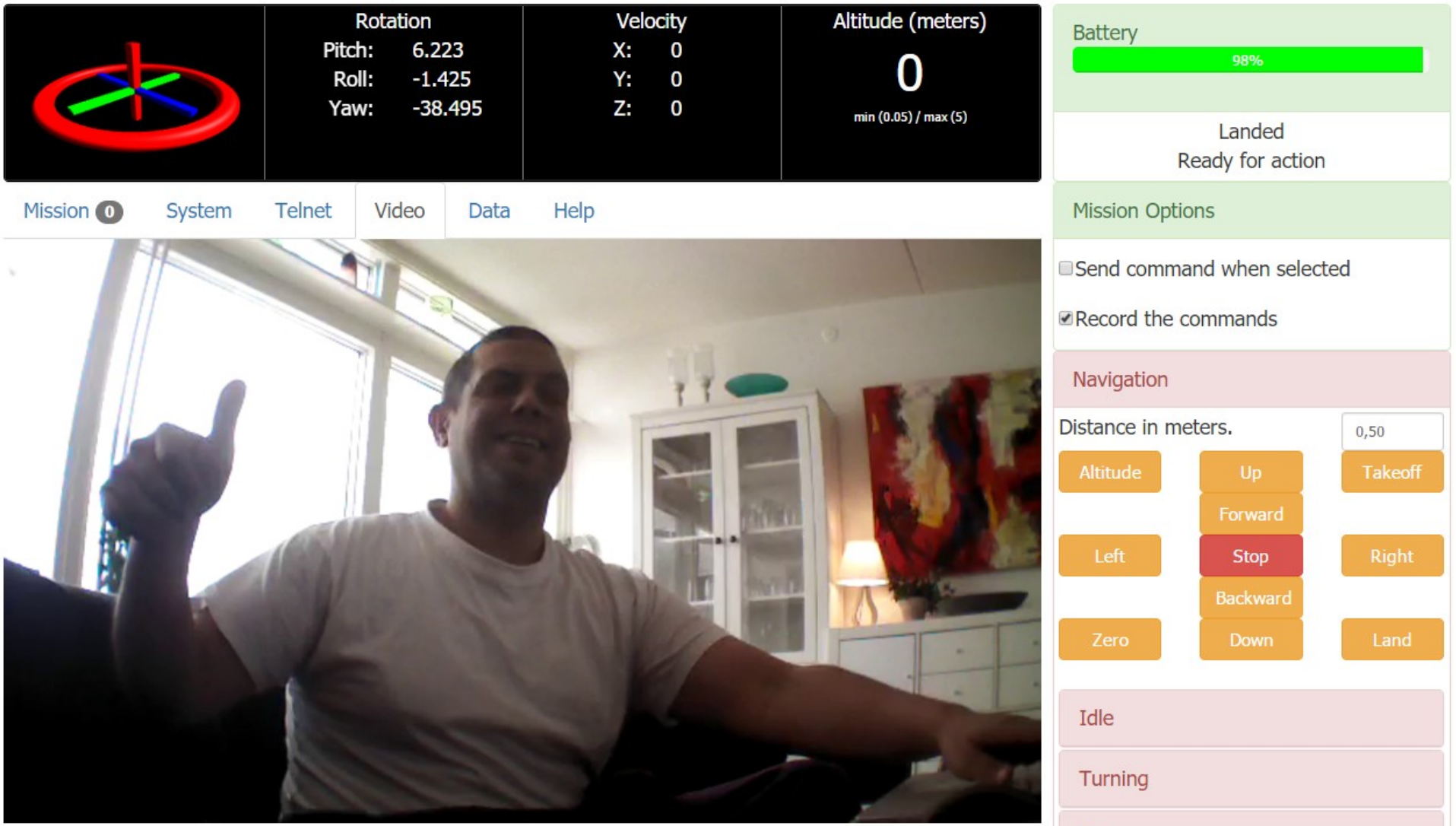
- Open a NodeJS console, in the folder: **<home folder>\node_modules\ar-drone\examples**
- Run **node simpleFlight.js**

MAKE SURE THE DRONE HAS ENOUGH SPACE TO TAKEOFF !!



A “HELLO WORLD” SNIPPET

```
var drone = require("ar-drone");
var client  = drone.createClient();

client.takeoff();
client.after(5000, function() {
  this.clockwise(0.5);

}) .after(3000, function() {
  this.stop();
  this.land();
});
```

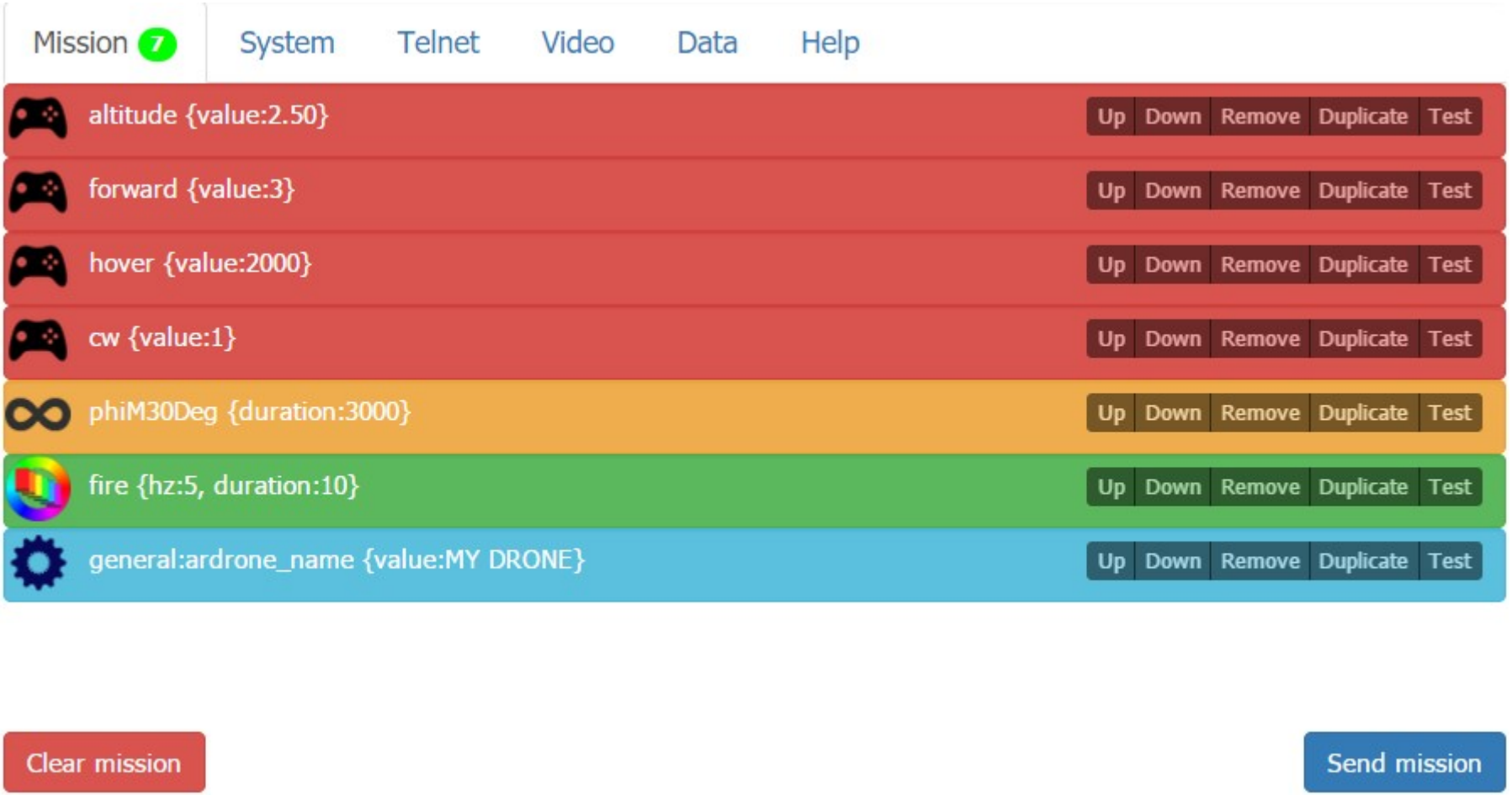


PROGRAMMING THE DRONE IS EASY!

Made this web based drone console, with NodeJS, JQuery, Angular, Web Sockets, WebGL..., in less than a week and I was totally in flow, just like when I played with Lego as a child.

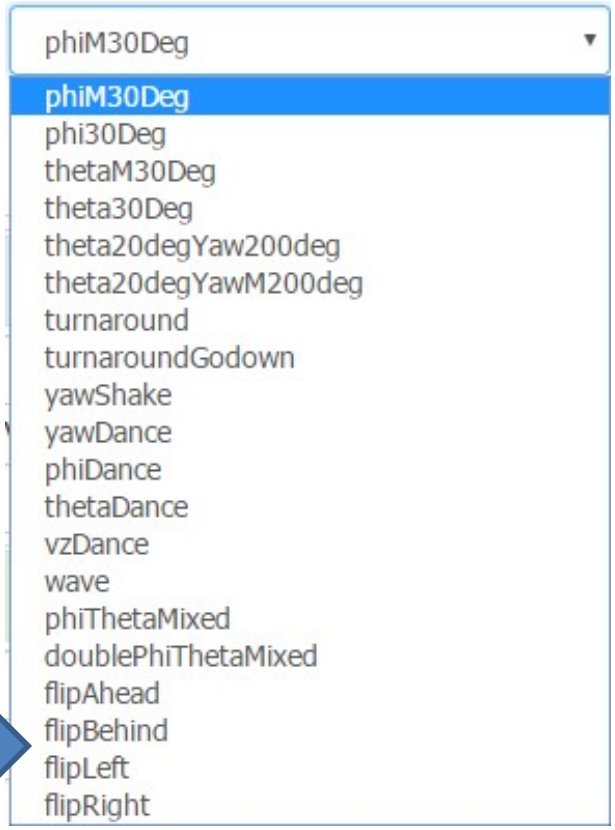
THE MISSION PANEL

Making it possible to define a mission queue and send a timed and specific navigation-, configuration- and LED mission to the drone, including waiting between commands.



THE NAVIGATION PANEL

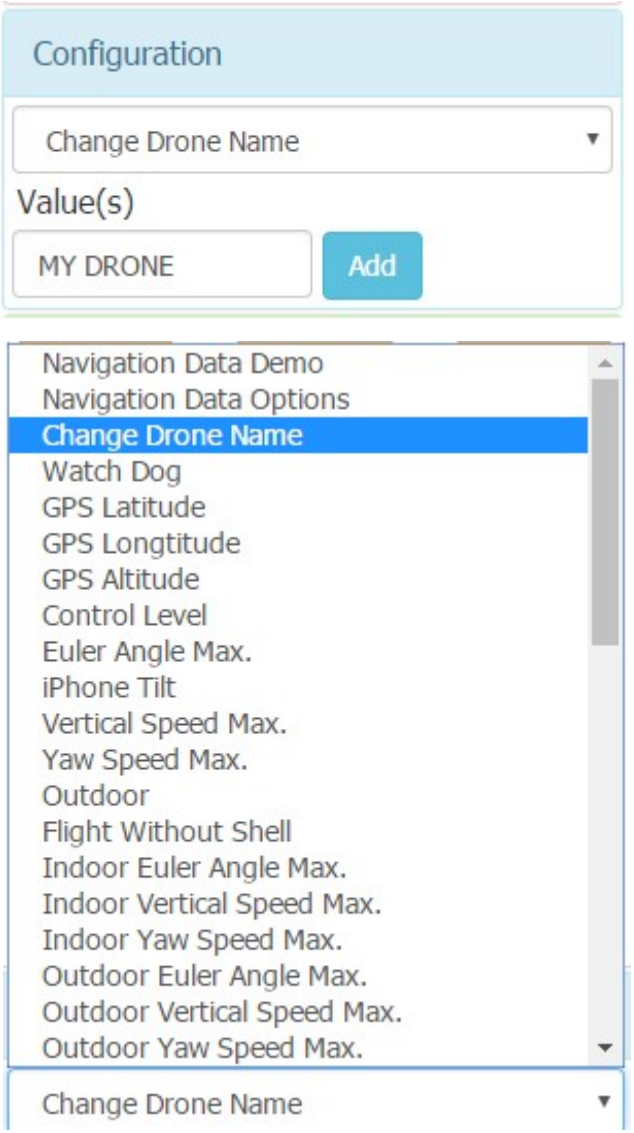
The drone has a set of cool animation gestures, which can be used via the API and the Freeflight mobile app.



THE CONFIGURATION PANEL

YOU CAN CONFIGURE LOTS OF SETTINGS FOR THE DRONE.

general:navdata_demo	control:control_level	gps:latitude
general:navdata_options	control:euler_angle_max	gps:latitude
general:ardrone_name	control:control_iphone_tilt	gps:altitude
general:com_watchdog	control:control_vz_max	
	control:control_yaw	network:ssid_single_player
leds:leds_anim	control:outdoor	network:ssid_multi_player
	control:flight_without_shell	network:wifi_rate
video:video_channel	control:indoor_euler_angle_max	network:owner_mac
video:codec_fps	control:indoor_control_vz_max	network:wifi_mode
video:video_codec	control:indoor_control_yaw	
video:video_slices	control:outdoor_euler_angle_max	detect:detections_select_h
video:video_live_socket	control:outdoor_control_vz_max	detect:detections_select_v
video:bitrate	control:outdoor_control_yaw	detect:detections_select_v_hsync
video:max_bitrate	control:flying_mode	detect:detect_type
video:bitrate_control_mode	control:flight_anim	detect:enemy_colors
video:bitrate_storage	control:hovering_range	detect:groundstripe_colors
video:video_on_usb	control:altitude_min	detect:enemy_without_shell
video:video_file_index	control:altitude_max	



THERE ARE MORE SETTINGS, ON THE DRONE FILE SYSTEM.

Via telnet you can access files like,

- cat /data/config.ini
- cat /proc/cpuinfo
- cat /proc/meminfo

And a lot of other files and settings.

NAVDATA

- The drone sends a variety of categorized and subscription-enabled data entities
- Here are some of the them
  - droneState, demo , rawMeasures, physMeasures, gyroOffsets, eularAngles, references, trims, pwm, altitude, visionRaw, vision, visionPerf, trackersSend, visionDetect, adcDataFrame, videoStream, games, pressureRaw, magneto, windSpeed, kalmanPressure, hdvideoStream, wifi, gps...

USING THE DRONE NAVDATA

```
// JS BACKEND: SUBSCRIBING TO NAVDATA FROM THE DRONE, ON THE SERVER SIDE.
var drone = require("ar-drone");
var constants = require("ar-drone/lib/constants");

var client  = drone.createClient();
var options = (
          1 << constants.options.DEMO) |
          1 << constants.options.VISION_DETECT |
          1 << constants.options.MAGNETO |
          1 << constants.options.WIFI |
          1 << constants.options.ALTITUDE |
          1 << constants.options.ZIMMO_3000
          );

client.config("general:navdata_options", options);
client.config("video:video_channel", 0);
client.config("detect:detect_type", 12);
```

```
// JS FRONTEND: SUBSCRIBE TO THE NAVDATA EMITS, FROM THE SERVER SOCKET.
client.on("navdata", function(data) {
  console.log("Flying:  " + data.droneState.flying);
  console.log("Battery %:  " + data.demo.batteryPercentage);
});
```

SAMPLE NAVDATA ENTITIES

droneState: {

```
    "flying": 0,
    "videoEnabled": 0,
    "visionEnabled": 0,
    "controlAlgorithm": 0,
    "altitudeContolAlgorithm": 1,
    "startButtonState": 0,
    "controlCommandAck": 0,
    "cameraReady": 1,
    "travellingEnabled": 0,
    "usbReady": 0,
    "navdataDemo": 0,
    "navdataBootstrap": 0,
    "motorProblem": 0,
    "communicationLost": 0,
    "softwareFault": 0,
    "lowBattery": 0,
    "userEmergencyLanding": 0,
    "timerElapsed": 0,
    "MagnometerNeedsCalibration": 0,
    "anglesOutOfRange": 0,
    "tooMuchWind": 0,
    "ultrasonicSensorDeaf": 0,
    "cutoutDetected": 0,
    "picVersionNumberOk": 1,
    "atCodecThreadOn": 1,
    "navdataThreadOn": 1,
    "videoThreadOn": 1,
    "acquisitionThreadOn": 1,
    "controlWatchdogDelay": 0,
    "adcWatchdogDelay": 0,
    "comWatchdogProblem": 0,
    "emergencyLanding": 0
}
```

rawMeasures: {

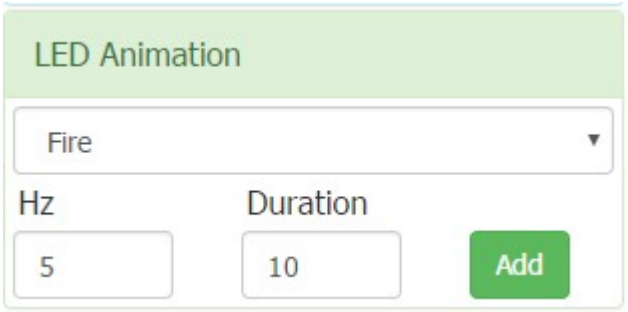
```
    "accelerometers": { "x": 2052,
                        "y": 2056,
                        "z": 2536},
    "gyroscopes": { "x": -46, "y": 8, "z": -9},
    "gyrometers": { "x": -46, "y": 8, "z": -9},
    "gyroscopes110":  { "x": 0, "y": 0},
    "gyrometers110": [0, 0],
    "batteryMilliVolt": 11632,
    "us": {
        "echo": { "start": 9302,
                  "end": 9559,
                  "association": 1,
                  "distance": 27},
        "curve": { "time": 8366,
                   "value": 0,
                   "ref": 120}
    },
    "usDebutEcho": 9302,
    "usFinEcho": 9559,
    "usAssociationEcho": 1,
    "usDistanceEcho": 27,
    "usCourbeTemps": 8366,
    "usCourbeValeur": 0,
    "usCourbeRef": 120,
    "echo": {       "flagIni": 1,
                    "num": 1,
                    "sum": 2490555 },
    "flagEchoIni": 1,
    "nbEcho": 1,
    "sumEcho": 2490555,
    "altTemp": 236,
    "altTempRaw": 236
}
```

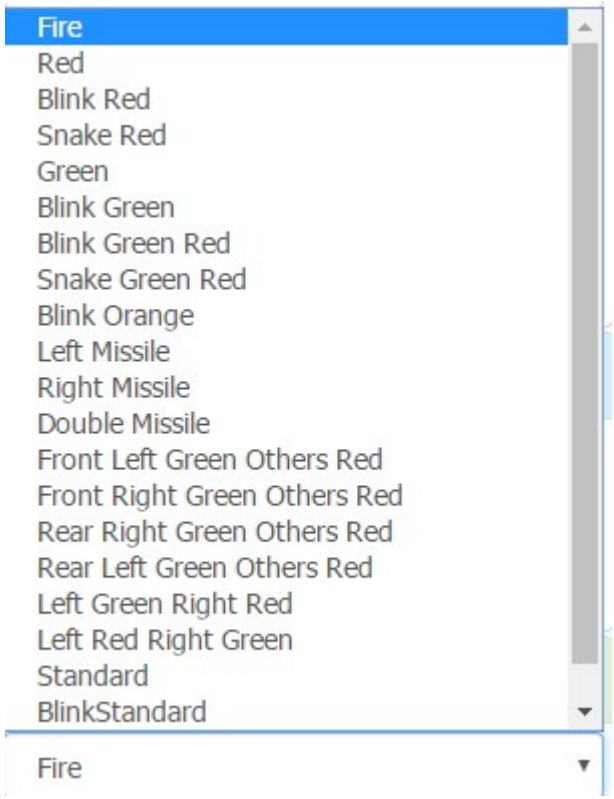
demo: {

```
    "controlState": "CTRL_LANDED",
    "flyState": "FLYING_OK",
    "batteryPercentage": 49,
    "rotation":  {
        "frontBack": -3.743,  "pitch": -3.743, "theta": -3.743, "y": -3.743,
        "leftRight": 1.913,  "roll": 1.913,  "phi": 1.913,  "x": 1.913,
        "clockwise": 73.349, "yaw": 73.349, "psi": 73.349, "z": 73.349
     },
    "frontBackDegrees": -3.743, "leftRightDegrees": 1.913,
    "clockwiseDegrees": 73.349, "altitude": 0, "altitudeMeters": 0,
    "velocity":  {"x": 0, "y": 0, "z": 0},
    "xVelocity": 0, "yVelocity": 0, "zVelocity": 0,
    "frameIndex": 0,
    "detection": {
        "camera": { "rotation": {"m11": 0, "m12": 0, "m13": 0, "m21": 0,
                                 "m22": 0, "m23": 0, "m31": 0, "m32": 0,
                                 "m33": 0 },
                    "translation":  {"x": 0, "y": 0, "z": 0 },
                    "type": 3 },
        "tagIndex": 0
    },
    drone":  {
    "camera":  {
          "rotation": {
          "m11": 0.2859284281730652, "m12": -0.9581588506698608,
          "m13": 0.013295086100697517, "m21": 0.9560242891311646,
          "m22": 0.28429126739501953,  "m23": -0.07208508998155594,
          "m31": 0.06528928130865097,  "m32": 0.03332161158323288,
          "m33": 0.9973098635673523
          },
          "translation": {  "x": 0, "y": 0, "z": -236}
    }
  }
}
```

THE LED ANIMATION PANEL

The drone comes with a set of LED light sequences, which can be used via the API.

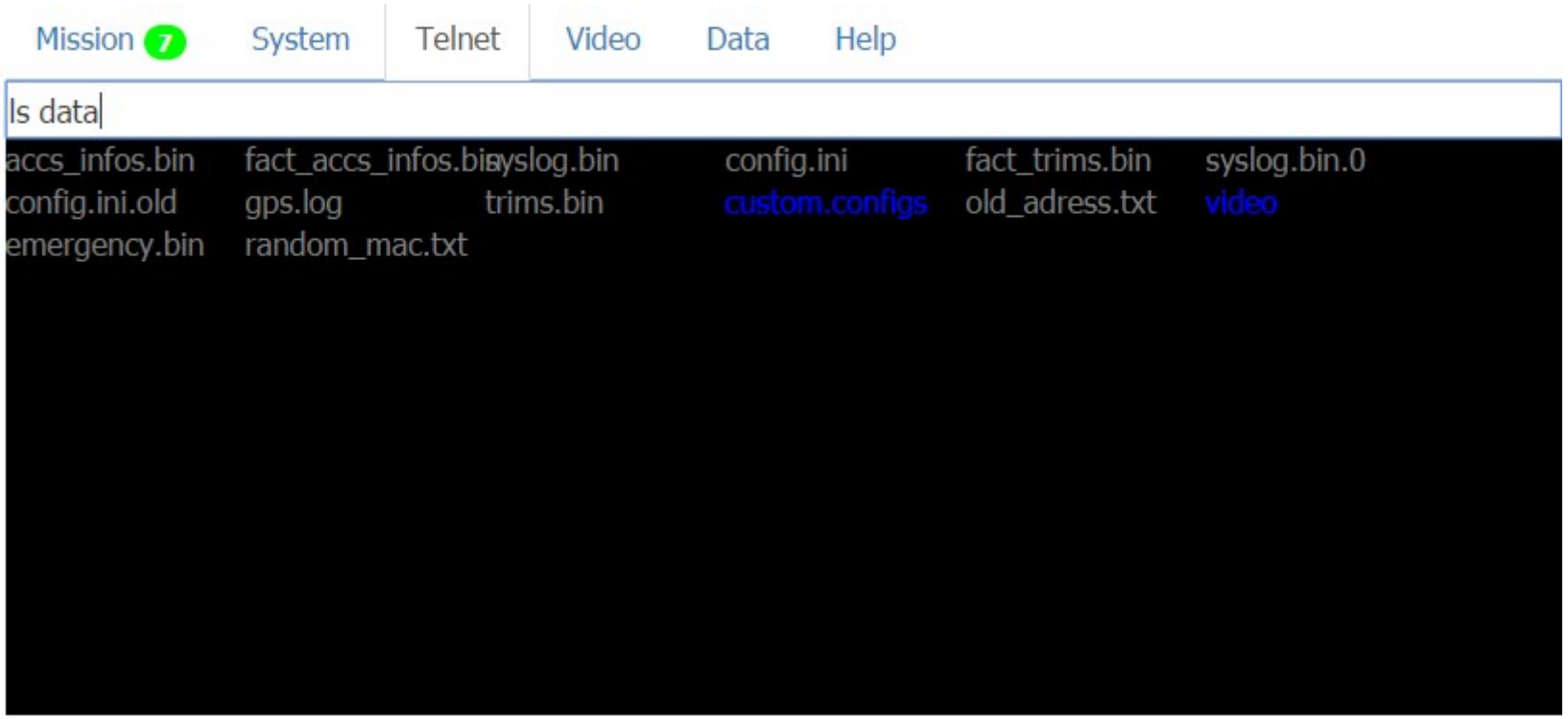




http:/gauth.fr/2011/09/control-the-ar-drone-leds/

THE TELNET PANEL

You have root access, via telnet, to the drone operating system, via the API.



ACCESS DRONE DATA VIA TELNET

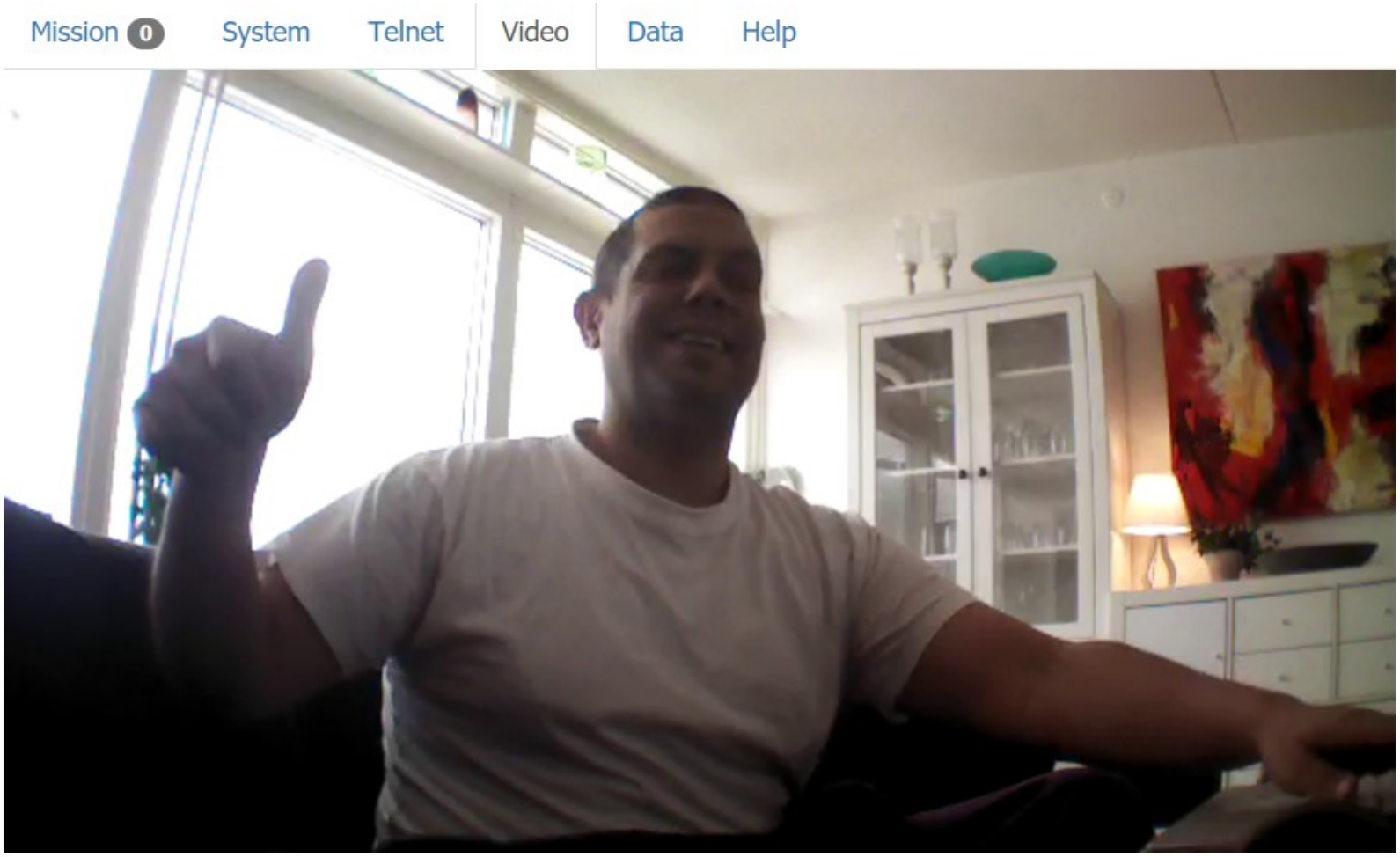
```
var telnetClient = require("telnet-client");
var telnetConnection = null;
var params = { "host" : "192.168.1.1", "port" : 23};
var data = {"config":{}, "cpu":{}, "mem":{}};

try {
  telnetConnection = new telnetClient();
  telnetConnection.connect(params).then( function(prompt) {
    telnetConnection.exec("cat /data/config.ini").then(function(res) {
      nvp2Json(res, DELIMITER_EQUALS, data.config);
      telnetConnection.exec("cat /proc/cpuinfo").then(function(res) {
        nvp2Json(res, DELIMITER_COMMA, data.cpu);
        telnetConnection.exec("cat /proc/meminfo").then(function(res) {
          nvp2Json(res, DELIMITER_COMMA, data.mem);
        })
      })
    });

  }, function(error) {
    console.log("Ups, failed to open telnet connection. ", error)
  });
} catch (error) { console.error("Ups, telnet operation failed. ", error); }
  finally { telnetConnection.destroy(); }
```

ACCESS DRONE DATA VIA FTP

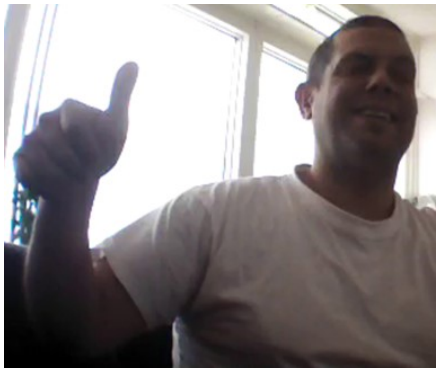
```
var JSFtp = require("jsftp");
var ftp = new JSFtp({ host: "192.168.1.1", port: 5551});
var data = {"config":{}, "cpu":{}, "mem":{}};


loadFtpData("/data/config.ini", DELIMITER_EQUALS, data.config);
loadFtpData("/proc/cpuinfo", DELIMITER_COMMA, data.cpu);
loadFtpData("/proc/meminfo", DELIMITER_COMMA, data.mem);


function loadFtpData(path, delimiter, target ) {
  var str = "";    // Will store the contents of the file.

  ftp.get(path, function(err, socket) {
    if (err) return;
    socket.on("data", function(d) { str += d.toString(); })
    socket.on("close", function(hadErr) {
      if (hadErr) { console.error("Failed to retrieving the file: " + path); }
      else { nvp2Json(str, delimiter, target); }
    });

    socket.resume();
  });
}
```

THE VIDEO PANEL

The drone comes with two cameras (front and bottom) which you can access and toggle via the API.



SHOWING THE VIDEO ON A WEB PAGE



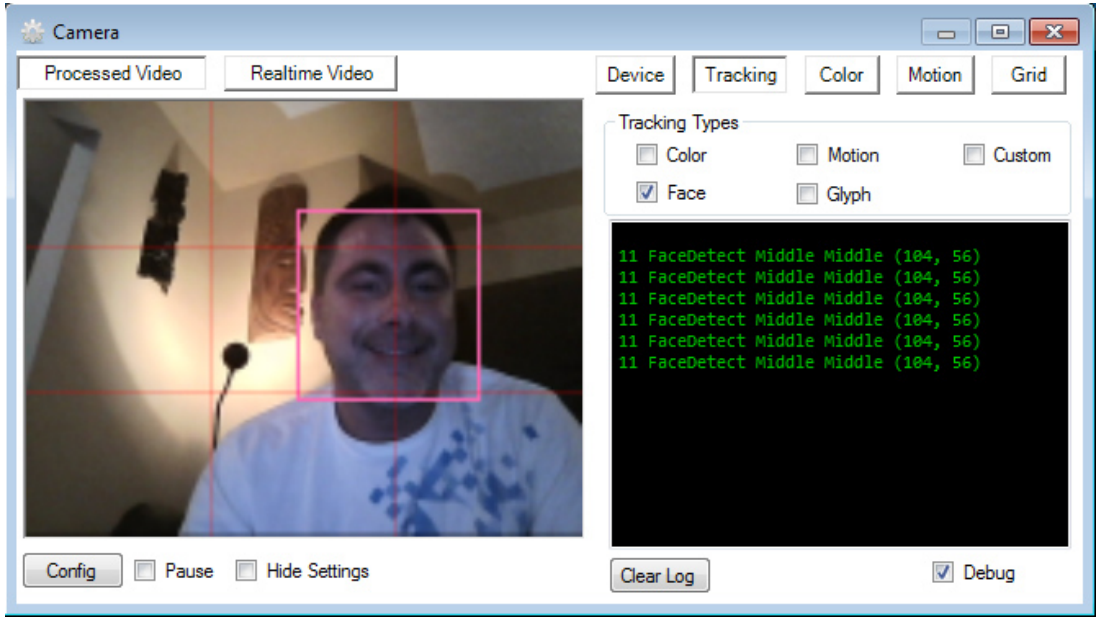
```
// PUT THIS <img id="video" …> TAG IN YOUR HTML.

var pngStream = client.getPngStream();
var urlCreator = window.URL || window.webkitURL;

pngStream.on('data', function (png) {
  var img = $("#video");
  var url = "public/no-video-available.png";

  try {
    var blob = new Blob([ png ], { type : "image/png" } );
    url = urlCreator.createObjectURL(blob);
  } catch (e) {
    console.log("Failed to handle video image.");
  }
  img.attr("src", url);
});
```

NOTE: YOU NEED THE LATEST FFMPEG VERSION, CONFIGURED AS A PATH SYSTEM VARIABLE.

Download and install ffmpeg.

IMAGE, OBJECT AND FACE RECOGNITION

- EZ-Builder enables you to make the drone react to movement of colors, faces, QR-codes, glyphs...
- You can train the drone to recognize new objects and patterns, check out these videos
  - https://www.youtube.com/watch?v=iVVLlag6eXU
  - https://www.youtube.com/watch?v=qqXhoJTn3AU



MISSION-BASED API

https://github.com/eschnou/ardrone-autonomy

- takeoff()
- land()
- stop()
- altitude(meters)
- up(meters)
- down(meters)
- left(meters)
- forward(meters)
- backward(meters)
- zero()
- go({x:0, y:0, z:0, altitude:0})
- cw(angle)
- ccw(angle)
- yaw(angle)
- hover(millis)
- after(millis, function)

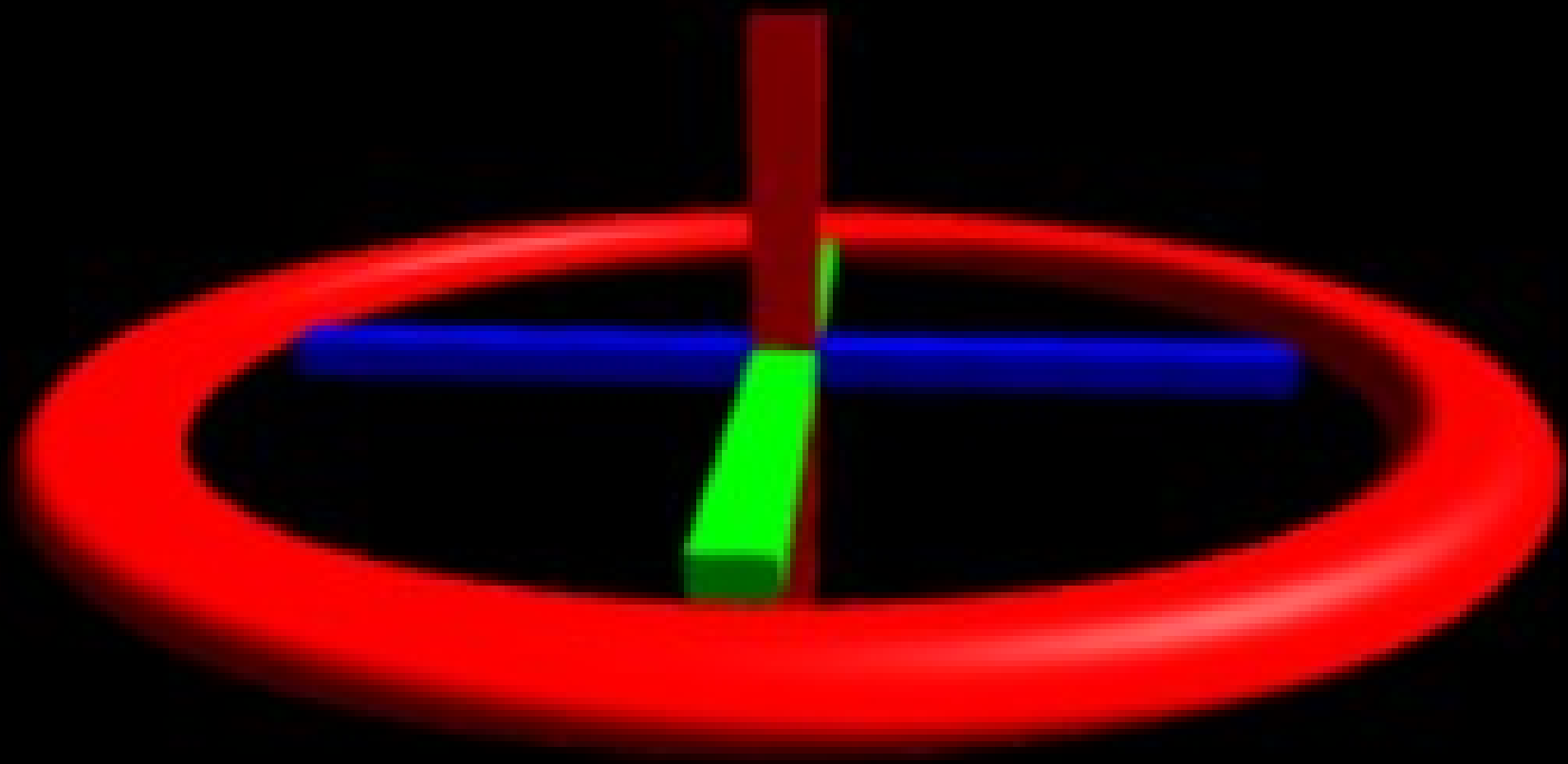
MISSION-BASED API

Flying in a 2 by 2 square pattern, at 2m in altitude and land.

```
autonomy = require('ardrone-atonomy');
mission  = autonomy.createMission();

mission.takeoff()
    .zero()
    .hover(500)
    .altitude(2)
    .forward(2).right(2).backward(2).go({ x: 0, y: 0 })
    .hover(500)
    .land();
```

https://github.com/eschnou/ardrone-autonomy

MAKING A WEBGL SCENE, FOR THE DRONE ROTATION (PITCH, ROLL & YAW)

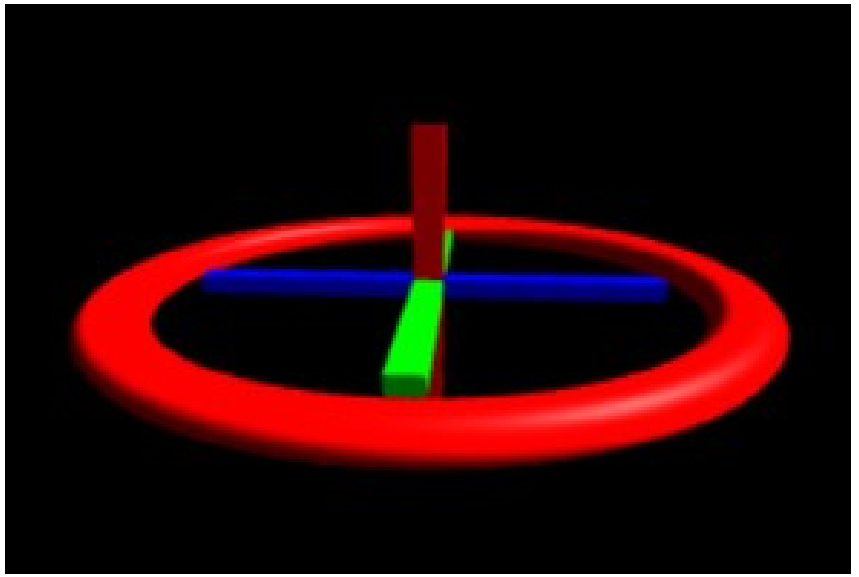


MAKING A WEBGL DRONE GYRO

With Three.js and WebGL activated in Chrome, you can make incredible 3D scenes with simple JavaScript. Here's the Three.js docs & samples.

Lets create a gyro-like animation, based on the drones rotational data.

```
<!DOCTYPE html>
<html>
   <head>
      <script src="…/three.min.js"></script>
   </head>
   <body>
      <div id="webglRotation"></div>
   </body>
</html>
```



CREATE SCENE, RENDERER & CAMERA

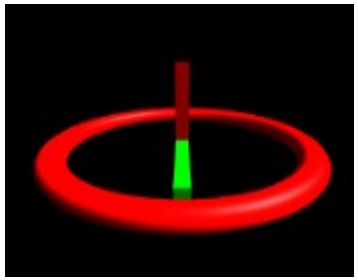
```
// WE NEED A SCENE, CAMERA AND RENDERER.
var scene = new THREE.Scene();
var renderer = new THREE.WebGLRenderer({ antialias : true});
renderer.setClearColor(0x0);


// HERE'S THE CAMERA.
var camera = new THREE.PerspectiveCamera(75, window.innerWidth /
                                              window.innerHeight, 0.1, 1000);
camera.position.z = 20;
camera.aspect = 1;
camera.updateProjectionMatrix();


// RELATE THE 3D RENDERER TO THE HTML DOM PLACEHOLDER OBJECT.
document.getElementById("webglRotation").appendChild(renderer.domElement);
```

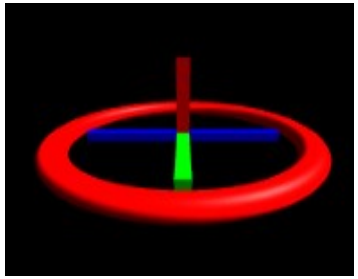
LET THERE BE LIGHT

```
// DIRECTIONAL SHINES FROM A DIRECTION AND ACTS LIKE THE SUN.
var dirLight = new THREE.DirectionalLight(0xffffff, 1);
dirLight.position.set(100, 100, 50);
scene.add(dirLight);


// AMBIENT GETS APPLIED TO ALL THE OBJECTS IN THE SCENE GLOBALLY.
var ambientLight = new THREE.AmbientLight(0x404040); // soft white light
scene.add(ambientLight);


// POINT AFFECTS OBJECTS USING MESHLAMBERTMATERIAL OR MESHPHONGMATERIAL.
var pointLight = new THREE.PointLight(0xffffff);
pointLight.position.set(50, 50, 50);
scene.add(pointLight);
```

MAKING A DRONE GYRO WITH WEBGL

```
var group = new THREE.Object3D();

// THE THORUS BASE.
var torus = new THREE.TorusGeometry(10, 1, 16, 100);
group.add( new THREE.Mesh(torus,
           new THREE.MeshPhongMaterial({ color: 0xff0000})));

// THE RED VERTICAL YAW LINE.
boxYaw = new THREE.BoxGeometry(1, 1, 16);
group.add( new THREE.Mesh(boxYaw,
           new THREE.MeshPhongMaterial({ color: 0xff0000})));

// THE GREEN HORIZONTAL PITCH LINE.
boxPitch = new THREE.BoxGeometry(1, 16, 1);
group.add( new THREE.Mesh(boxPitch,
           new THREE.MeshPhongMaterial({ color: 0x00ff00})));

// THE BLUE HORIZONTAL ROLL LINE.
boxRoll = new THREE.BoxGeometry(16, 1, 1);
group.add( new THREE.Mesh(boxRoll,
           new THREE.MeshPhongMaterial({ color: 0x0000ff})));
scene.add(group);
```





SCENE RENDERING

```
var render = function() {
  requestAnimationFrame(render);

  group.rotation.x = 90 + (pitch / 45.0);
  group.rotation.y = - (roll / 45.0);
  group.rotation.z = - (yaw / 45.0);

  renderer.render(scene, camera);
}};

setTimeout(render, 100);
```

L&B DRONE KODE TEST

http://www.youtube.com/watch?v=-mPb0NvKhfI

QUESTIONS

WHO WOULD....

like to code a drone and get in flow, a.s.a.p?

love to participate, at a drone code event day?

accept an entry fee, for drone equipment and dinner, at such an event? How much would you pay, max?

love to try a reel customer job, coding a drone app?

How do you see drone- and robotics technology merge & integrate with IT software paradigms and developer profiles?

THAT’S IT.
THANK YOU FOR YOUR TIME & ATTENTION.

If you have any feedback or questions,
please send it to ronni.kahalani@gmail.com

APPENDIX

DRONE APIS AND FRAMEWORKS

C / C# / C++

- MavLink - Micro Air Vehicle Communication Protocol
- ROS - Robot Operating System
- AutonomyLab/ardrone_autonomy (doc)

Java

- JavaDrone
- DroneKit.io (Android, Phyton, Cloud, ~~iOS~~)

JavaScript / NodeJS

- ar-drone
- ardrone-atonomy

THE SOFTWARE FOR TODAY

NodeJS

- https://nodejs.org/en/download

FFMPEG

- https://ffmpeg.org/download.html

PuTTYtel

- http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

EZ-Builder (EZ-Robots)

- https://www.ez-robot.com/EZ-Builder

DRONE SYSTEM LINKS

- http://busybox.net
- http://drone-apps.com/support/ar-drone-firmware
- http://wiki.paparazziuav.org/wiki/AR_Drone_2/AT_Commands
- https://parrotcontact.parrot.com/website/user-guides/download-user-guides.php?pdf=ar-drone-2/AR-Drone-2_User-guide_Android_UK.pdf

ARDRONE C-API GUIDE & HEADER CONSTANTS, USED BY AR-DRONE NODEJS API

- http://github.com/johnboiles/ARDroneSDK/blob/master/ARDroneLib/Soft/Common/ardrone_api.h
- http://web.mit.edu/tinali/www/ARDrone_Developer_Guide.pdf

AR-DRONE LINKS

- http://github.com/felixge/node-ar-drone

NODECOPTER

- http://www.nodecopter.com
- Nodecopter YouTube videos

FELIX GEISENDOERFER SITE & VIDEOER (AR-DRONE NODEJS & NODECOPTER FOUNDER)

- http://felixge.de
- https://vimeo.com/68725851
- https://www.youtube.com/watch?v=jl5v3bsMH_E
- https://www.youtube.com/watch?v=nwGNNMJt4mE&t=19m52
- https://www.youtube.com/watch?v=zT_FtkXZRWo

AR-DRONE LINKS

MULTIPLE DRONE NETWORK

- http://github.com/AutonomyLab/ardrone_autonomy/wiki/Multiple-AR-Drones
- http://www.robotappstore.com/Knowledge-Base/How-to-Program-ARDrone-Remotely-Over-WIFI/96.html

DRONE ATONOMY

- http://amitdesai03.blogspot.dk/2014/02/hack-for-autonomous-delivery-using.html

COMPUTER VISION, CAMERA MOTION TRACKING AND COLOR, FACE AND OBJECT RECOGNITION

- http://stackoverflow.com/questions/34713594/tracking-objects-from-camera-pid-controlling-parrot-ar-drone-2?rq=1
- http://github.com/taghof/Navigation-for-Robots-with-WIFI-and-CV
- http://www.pyimagesearch.com/2015/05/04/target-acquired-finding-targets-in-drone-and-quadcopter-video-streams-using-python-and-opencv
- https://github.com/bkw/node-dronestream
- https://github.com/tylerswhitfield/drone-bull

GPS

- http://stackoverflow.com/questions/24170884/get-gps-data-from-ar-drone-2-0

AR-DRONE LINKS

DRONE MOTION DEVICES

- http://charliegerard.wordpress.com/2015/01/20/drone-leap-motion-cylon-js
- https://github.com/TooTallNate/node-drone-joystick
- http://robohub.org/up-and-flying-with-the-ar-drone-and-ros-joystick-control

ROS - ROBOT OPERATING SYSTEM

- http://blogdugas.net/blog/2015/05/12/ar-dot-drone-2-with-ros-and-opencv-get-started-quick-with-ubuntu-or-mint

MAVLINK - MICRO AIR VEHICLE COMMUNICATION PROTOCOL

- http://diydrones.com/group/arducopterusergroup/forum/topics/mavlink-tutorial-for-absolute-dummies-part-i
- http://api.ning.com/files/i*tFWQTF2R*7Mmw7hksAU-u9IABKNDO9apguOiSOCfvi2znk1tXhur0Bt00jTOldFvob-Sczg3*lDcgChG26QaHZpzEcISM5/MAVLINK_FOR_DUMMIESPart1_v.1.1.pdf

OTHER DRONE SITES

- http://dronehacks.com , http://dronegames.co , https://github.com/wiseman/ar-drone-rest