## Welcome

Welcome to the first guide/tutorial on flying the Parrot AR 2.0 drone! These guides are based on my own experience of learning and working with the AR 2.0 drone so I’ll try to write these in a way that I would’ve found useful and easily understandable when I was getting started.

Also, as many people (not just computer science students/grads, like me) enjoy flying drones, I’ll do my very best to assume as little programming knowledge as possible. Undoubtedly there will be some bits I accidentally gloss over so please feel free to comment on this post if there are any gaps you’d like covered.

## Stuff you’ll need

Before we get in to anything, these are the things you’ll need to get going with this guide.

### 1. Windows 10 PC/laptop with Wi-Fi

For these tutorials, I am using a Windows 10 PC with a USB Wi-Fi adapter to connect to the drone, if you don’t already have Wi-Fi capability on your PC, something like the USB adapter in the link below should solve that for you.

<https://www.amazon.co.uk/TP-LINK-TL-WN725N-150Mbps-Wireless-N-Adapter/dp/B008IFXQFU/ref=sr_1_2?s=computers&ie=UTF8&qid=1475758246&sr=1-2&keywords=usb+wifi>

If you have a Windows 10 laptop, you should be all set (built-in Wi-Fi) ☺

**Windows:** Other versions of Windows should be fine, I initially completed some of this work on a Windows 7 laptop so that should be fine, Windows 8 (and 8.1) are notoriously awkward operating systems however so there may be a need to Google some bits and pieces if you run into issues. Feel free to comment on this post with any issues as well though!

**Linux/Mac:** At the moment I’m looking into getting a MacBook of my own so one day I should be able to try this all out on Mac and write a guide on that if needed. Might as well wait to do Linux until that point as well!

### 2. A drone!

Aside from that, you’ll also need a Parrot AR 2.0! You can find these on Amazon for under £200 brand new, but check the used/new options from other sellers as I managed to get it even cheaper through Amazon Warehouse Deals, just with a damaged box! The link below should give you an indication of the current cost of these drones from Amazon.

<https://www.amazon.co.uk/Parrot-Drone-Elite-Quadricopter-Sand/dp/B00FS7SSD6/ref=sr_1_3?ie=UTF8&qid=1476196262&sr=8-3&keywords=parrot+ar+drone+2.0>

### 3. Spare battery (optional, but recommended)

I would definitely recommend getting hold of an additional battery for the drone as the flight time on the standard battery (1000mAh Li-ion polymer) is around 10-12 minutes, give or take. I got hold of a 1500mAh battery to use while the standard one is charging (and vice-versa) which worked out quite nicely. The 1500mAh I bought is linked below and will allow for an absolutely breath-taking 18 minutes of flight time. This will cost around £40 which seems like a bit of a sting to the wallet but it with save you twiddling your thumbs for around 45 minutes while you’re waiting for a battery to charge before you can fly again!

<https://www.amazon.co.uk/Parrot-AR-Drone-1500mAh-Lithium-Polymer/dp/B00DAL5GD2/ref=sr_1_2?ie=UTF8&qid=1476196958&sr=8-2&keywords=parrot+ar+2.0+battery>

Aside from that I would recommend downloading Notepad++ as we’ll be fiddling with some code in JavaScript. Notepad++ is a nice, basic text editor that also understands how code should be formatted and highlights elements of code, which will really help anyone get to grips with basic coding. You can get hold of Notepad++ from the link below.

<https://notepad-plus-plus.org/download/>

## Intro

In this guide we are going to do the following:

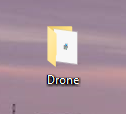
1. Install an application that will allow us to send and receive information from our drone (NodeJS)
2. Connect to the drone using Wi-Fi
3. Use NodeJS to tell the drone to take off and land from our laptop
4. Write some simple JavaScript programs to save us typing lots of commands each time we want to fly the drone
5. Add some more functionality into our JavaScript programs and understanding the use of these to control the pitch, roll and yaw movements of the drone

To complete this, I have used a fantastic guide from the Instructables website (link below). This contains many of the commands that we’ll use, however I’ve tailored this guide to fill in any gaps I found in the Instructables guide and to make sure it works with the latest Windows operating system.

<http://www.instructables.com/id/Autonomous-AR-Parrot-Drone-20-Flying/step2/Node-JS/>

## Install NodeJs

Just before we get into installing NodeJS, the first thing I did was create a folder on the Desktop called ‘Drone’ which contained an empty folder called ‘nodejs’. When we install NodeJS below we’ll change the default install location from C:\Program Files\nodejs\ to C:\Users\Mark\Desktop\Drone (just change ‘Mark’ to the user name on your PC!)

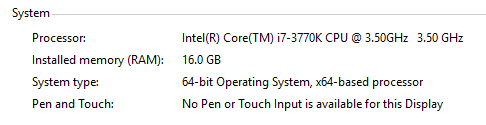


Now we’ve got a place sorted out for NodeJS to live, click the following link to view the available NodeJS downloads <https://nodejs.org/en/download/>.

The version you download will ultimately depend on the operating system you are running, in Windows you can check this in the following ways:

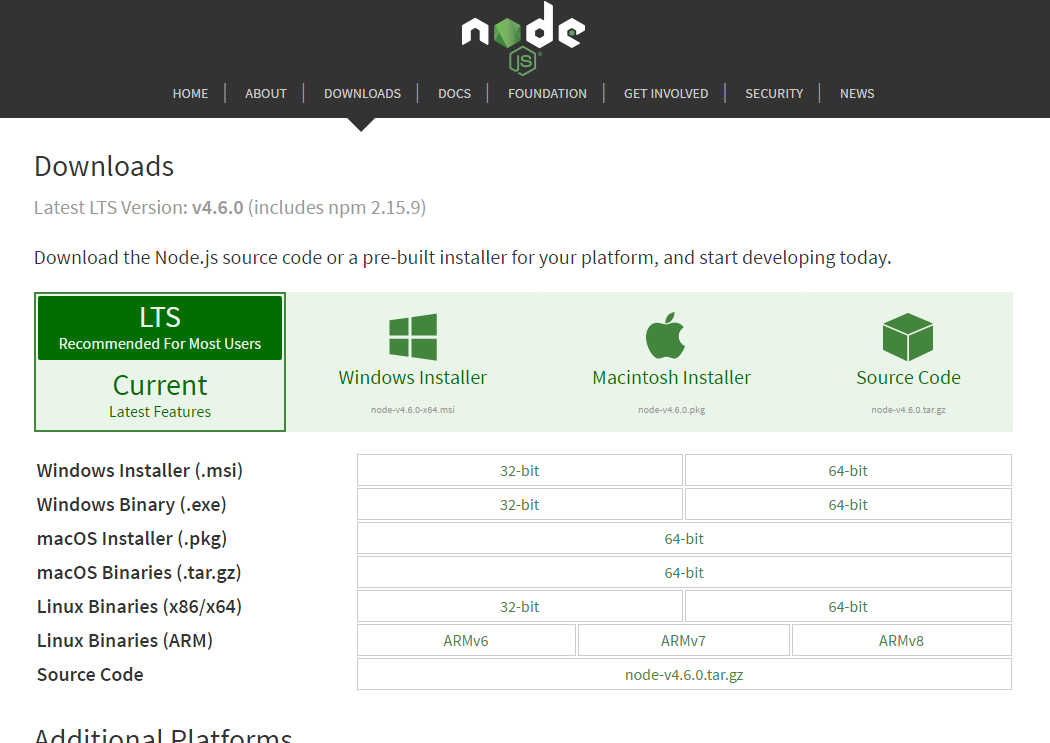
* Windows 7 – right click the ‘Computer’ option in the start menu and click the ‘Properties’ option
* Windows 8 and 10 – hold down the windows key (key in-between CTRL and ALT on the bottom left of keyboard) and press E to open file explorer, then right click ‘This PC’ on the left hand side of the file explorer window and click ‘Properties’

The resulting window should contain the ‘System’ section as shown below.



According to the result given above for my PC, I’m running a 64-bit operating system (‘System type’ field) so I’ll need to download the 64-bit version of NodeJS (use the .msi installer) as shown by the red box in the next screenshot.

**Note:**  Although it is possible to run 32-bit versions of software on 64-bit operating systems, it’s best practice to use the most appropriate install for your machine as using a 32-bit program on a 64-bit operating system can waste some system resources. In hobbyist projects this isn’t a significant issue but it’s a good practice to follow if your technological adventures lead to bigger and better places!



DEFAULT Installs to C:\Program Files\nodejs\

*I cocked up here, installed it into the default location above and tried running it in my own drone folder on the desktop. It had no idea what was going on so uninstalled node completely and reinstalled in it’s own folder within the drone folder (Desktop/drone/nodejs)*

*MY install location C:\Users\Mark\Desktop\Drone\nodejs\*

Ran npm install ar-drone to get hold of Felixge’s node library for controlling the drone

* hold left-shift and right click inside nodejs folder to be given the option to open command window in that location
* now type npm install ar-drone as the guide says
* ignore warnings that pop up in command line window

## Connect to Drone

Connect to drone wifi (ardrone2\_062272 in my case), I used a USB wifi adapter for this (using desktop computer rather than laptop this time around!) This sort of thing should do the trick:

<https://www.amazon.co.uk/TP-LINK-TL-WN725N-150Mbps-Wireless-N-Adapter/dp/B008IFXQFU/ref=sr_1_2?s=computers&ie=UTF8&qid=1475758246&sr=1-2&keywords=usb+wifi>

When connected by Wifi – it will say ‘No internet, open’ regarding the connection to your drone, if on Win10 like me. This is fine.

## Running NodeJS and manually controlling drone (takeoff and land)

Once on Wifi of drone, locate node program on your computer, ‘…\nodejs\’ as mentioned above, double-click green/grey node.exe icon to open the NodeJS terminal window (similar to windows command line.

Node will be your means of sending commands to the drone and receiving any outputs back (such as video feed using ffmpeg which will be described later).

Type in the commands shown in the instructables guide one by one, making sure each line ends with a semi-colon ‘;’ pressing enter before the next line. **Note:** copy the final (‘client.land();’) command before pressing enter on the takeoff command! I.e. copy and paste each command line by line from the guide into the nodejs terminal window using right-click then copy, right-click then paste making sure to copy the final ‘land’ command before pressing enter on the takeoff command, just in case you need to stop the drop quickly without risking your fingers!

As the instructable guide says, you should now have the ability to let the drone hover a few feet off the floor until you enter the ’client.land();’ command.

## Running commands as a simple, repeatable JavaScript program instead

For this I installed Notepad++ as this is a fantastic, free and simple text editor (like notepad), that comes equipped with code assistance which will help us visualise the structure of our simple program.

Notepad++ <https://notepad-plus-plus.org/download/v7.html>

Enter the lines as shown in Step 5 of the instructables guide, apart from line 5 ‘client.land();’. The author accidentally kept this line in, running this program without removing this line will result in the drone being told to takeoff and land straight away without any time inbetween (not very exciting!). Enter the program as shown below then save the file as something like firstflight.js in the same folder as your node.exe application (…/nodejs).

var arDrone = require('ar-drone');

var client = arDrone.createClient();

client.takeoff();

client

.after(5000, function() {

this.land();

});

Now to run this program you need to open the windows command prompt in the location of your firstflight.js file and enter the following command – *node firstflight.js*

**Note:** if your drone just sits on the ground with its lights flashing, it is likely you’ve got a nodejs window open somewhere, make sure all nodejs windows are closed! The above command calls node and tells it to run a particular program - n*ode firstflight.js* essentially means “computer, I want you to use a program called node [node.exe as we know it] to run a JavaScript program called firstflight.js, simple as that!

**Another note:** I asked you to open command prompt in the directory of the firstflight.js program you wrote, this isn’t mandatory, it just saves you having to type out the whole path to your program like *node C:\Users\Mark\Desktop\Drone\nodejs\firstflight.js* – that would be a pain to enter every time compared to *node firstflight.js*!

## Adding more commands in JavaScript

Made a second JS file called secondflight.js (copied and renamed firstflight.js). Now going to fiddle with the following commands as described in the Instructables guide:

* up(speed) - has the drone gain altitude at a speed between 1 (max speed) and 0 (still).
* down(speed) - makes the drone reduce altitude
* clockwise(speed) - drone spins clockwise
* counterClockwise(speed) - drone spins counter-clockwise
* front(speed)/back(speed) - changes the pitch causing horizontal movement
* left(speed)/right(speed) - changes the roll causing horizontal movement
* stop() - keeps the drone hovering in place

The file secondflight.js demonstrates adding a bit more functionality to our drone program. In this file we’re telling the drone to take off, go forward a bit (front) at 20% (0.2) of maximum speed, then stop after 2 seconds of going forward (lines 10 and 11, .after(2000, function() {this.stop()}. The same is then repeated but instead of going forward again (front), we’re telling it to go backwards (back), before stopping and then landing.

That was pretty straightforward, so just to concrete the idea in our minds about how these commands are built and work with the drone, thirdflight.js demonstrates the concept of ‘yaw’ using the clockwise and counterClockwise commands. Yaw is a movement where the drone will stay at the same height and rotate in mid-air without going forward or backwards – like putting a pen flat on its side on a table and rotate it, this is yaw! See the image below for a visual of yaw in terms of an aeroplane, linear rotation on the perpendicular axis of the object.



Figure 1: http://machinedesign.com/site-files/machinedesign.com/files/uploads/2014/06/PRY.gif