instructables

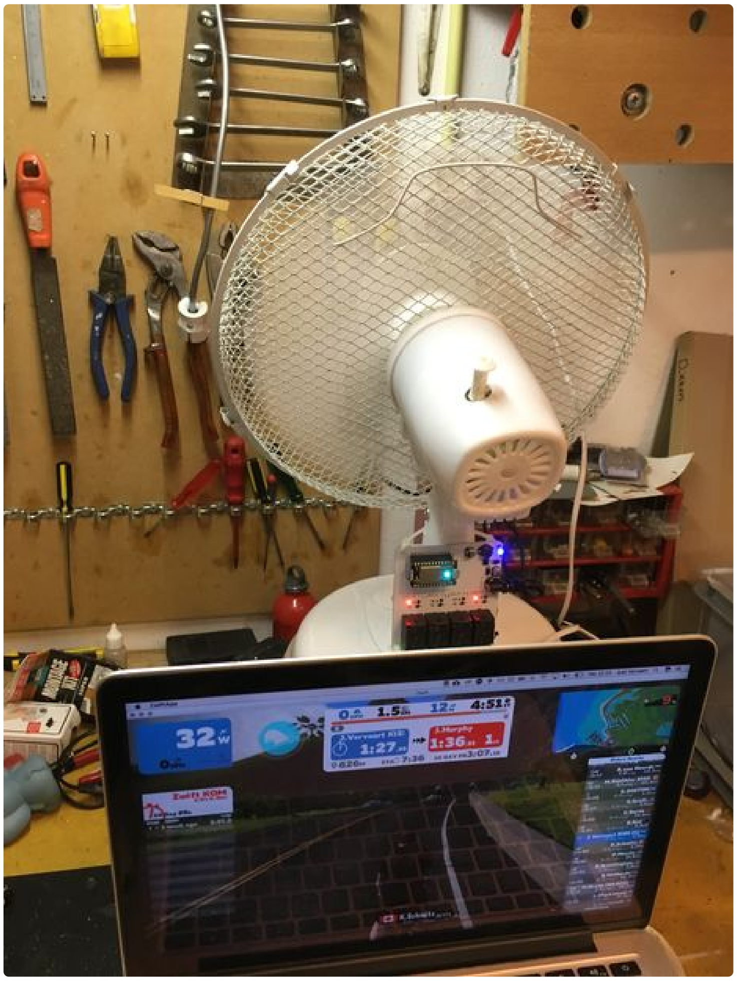
Internet Connected Fan for Use With Zwift

by justvervaart

I made a fan that's connected to the internet for use with Zwift, a virtual bike racing game / training system. When u go faster in Zwift, the fan turns faster to simulate outside riding conditions. ;) I had some good fun building this, hope u enjoy building this yourself.

! use these instructions at your own risk, your working with deadly currents, so be careful !

https://vimeo.com/249658866



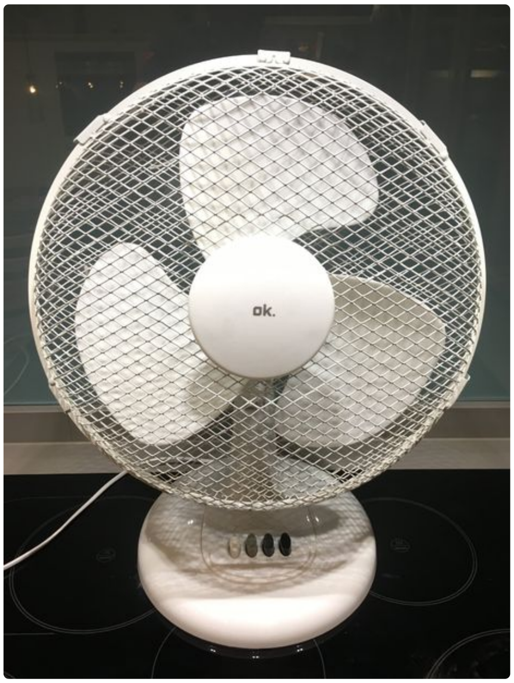
Step 1: Parts & Tools

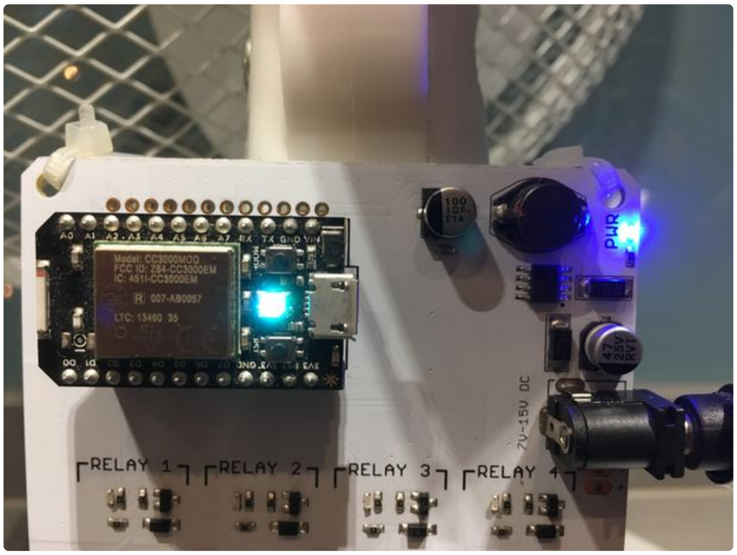
First you need a fan. I had a fan lying around with 3 different speeds, so that's what i used. If you have one with 2 or 4 speeds you can simply adjust the code that's controlling the fan. But it has to be a fan with buttons. I found this one Amazon. And because i'm Dutch, here is a link to a fan on bol.com that would work. It's going to cost you about $30,-

Then we need a device to connect the fan to the internet and something to control it with. I used a Photon from Particle. The make easy to program IoT devices. The Photon cost $19,- We also need a Relay Shield to control the fan. I used an older model, so it looks a little different, but the new model should work just fine with. Cost $30.- You also need an DC adapter to power the relay shield.

You also need a computer to run some scripts while using the fan. Because I just Zwift on my macbook, this is what I used building this, so these instructions are for use with a Mac. But if you are good friends with your Windows machine I think it should be possible to get it to work on such a device as well. And if your really nifty you could probably make the script run on a server (or in the cloud), if you do so, please let me now.

Finally, we need some tie-raps, short pieces of wire for high currents, screwdriver(s) en a pair of wirecutters.





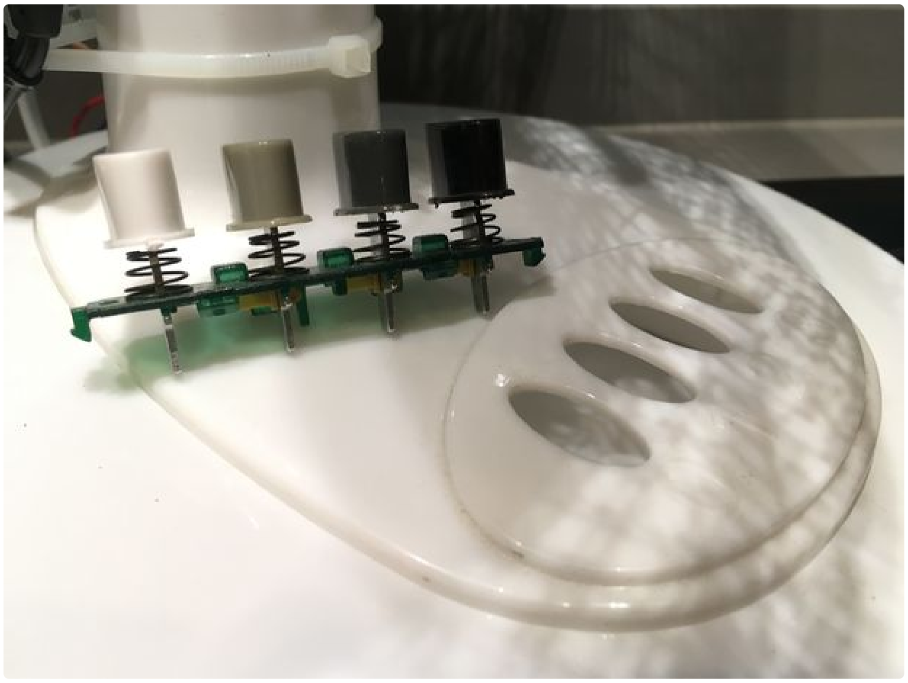
Step 2: Hacking the Fan

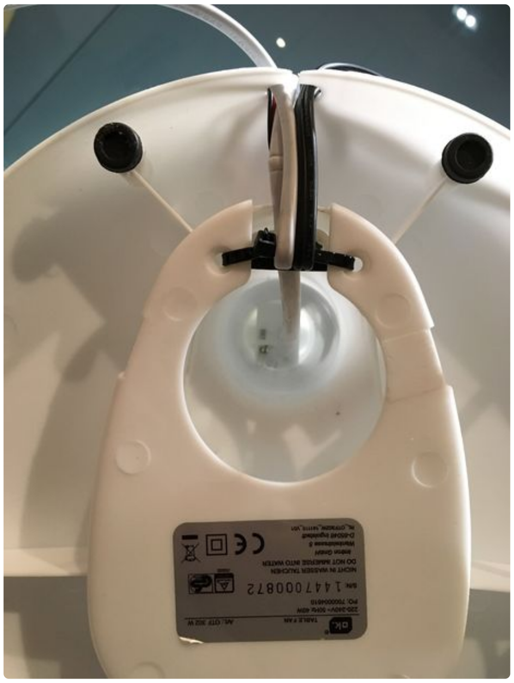
Open the fan *(remove the plug first ;)* and be sure to write down the colors of the wires associated with the different speeds (1, 2 & 3) before u begin the next step. Note that there is also a wire connected to the button housing not connected to one of the buttons. This is the wire feeding the power (common). Remove the buttons from the housing and disconnect all the wires.

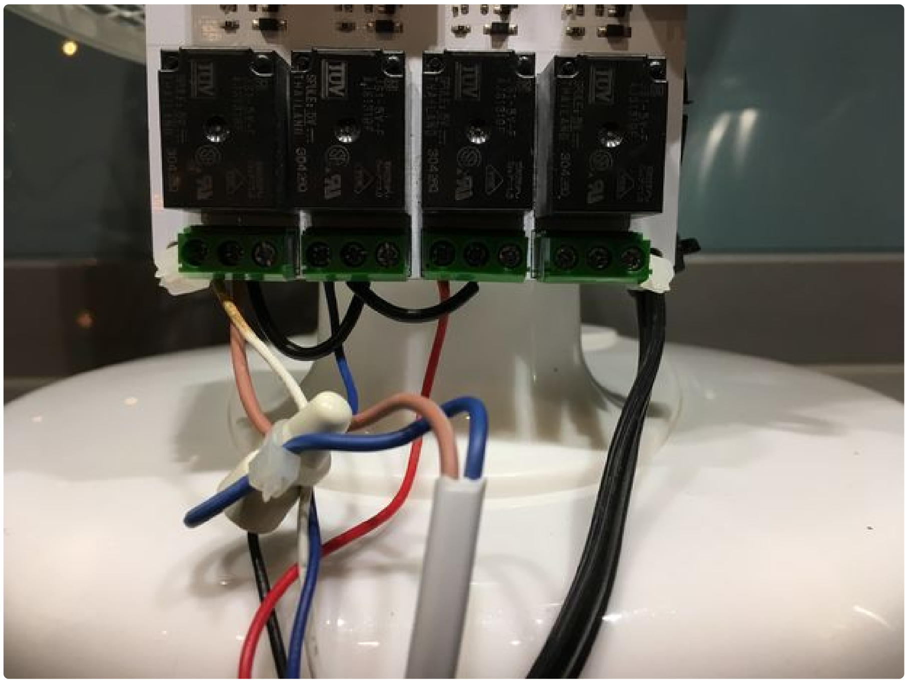
Notice that each relay has 3 connections to use. NO, NC and COMM. NO stands for Normally Open, NC means Normally Closed en COMM for Common. We want to connect the fan to the NO so nothing happens until we want it to. Connect the wire for speed 1 to the NO on relay 1, the wire for speed 2 to relay 2, and wire 3 on relay 3.

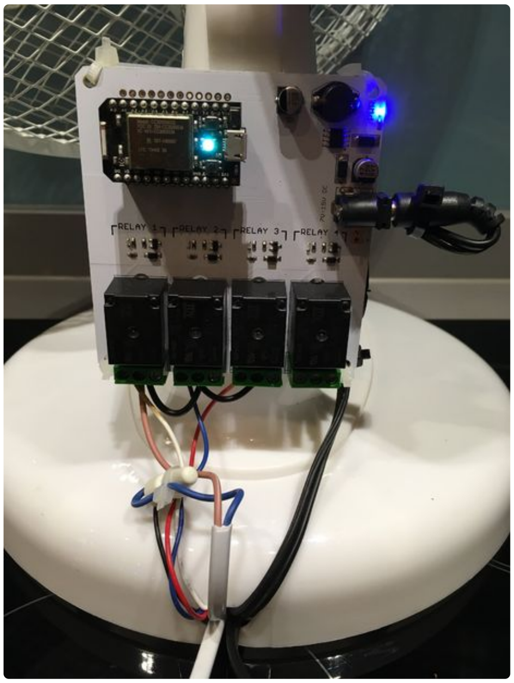
Then connect the common wire to COMM on relay 1 and make a connection from COMM on relay 1 to COMM on relay 2 with a short piece of wire (suitable for 220v) and also from COMM on relay 2 to COMM on relay 3.

I connected the relay shield to the base of the fan with some tie-raps for demo purposes. Best would be to build a housing, because of exposed contact with 220v on them! **Please be careful, especially with kids around!**







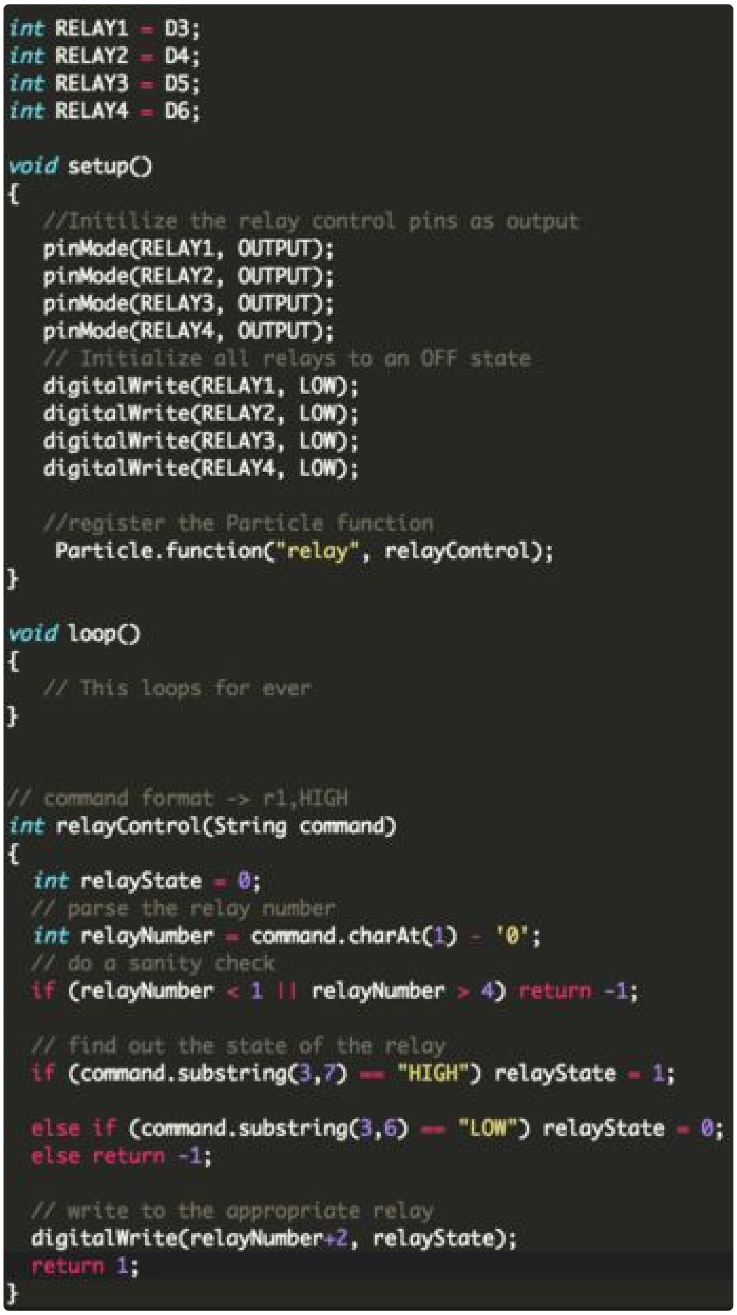


Step 3: Put the Code on the Photon

First: connect the Photon to the same wifi network as your computer. With the Photon devices comes a complete set of instructions on how to do this.

Second: the Photon needs to run some code to be able to control the relay shield. Take the code from the attached file and load it on your Photon. This code makes it possible tot control the relays via the internet. It is not necessary to edit this code.

If needed, there is a great community to help you out if you get stuck!



```
int RELAY1 = D3;
int RELAY2 = D4;
int RELAY3 = D5;
int RELAY4 = D6;

void setup()
{
   //Initilize the relay control pins as output
   pinMode(RELAY1, OUTPUT);
   pinMode(RELAY2, OUTPUT);
   pinMode(RELAY3, OUTPUT);
   pinMode(RELAY4, OUTPUT);
   // Initialize all relays to an OFF state
   digitalWrite(RELAY1, LOW);
   digitalWrite(RELAY2, LOW);
   digitalWrite(RELAY3, LOW);
   digitalWrite(RELAY4, LOW);

   //register the Particle function
    Particle.function("relay", relayControl);
}

void loop()
{
   // This loops for ever
}


// command format -> r1,HIGH
int relayControl(String command)
{
  int relayState = 0;
  // parse the relay number
  int relayNumber = command.charAt(1) - '0';
  // do a sanity check
  if (relayNumber < 1 || relayNumber > 4) return -1;

  // find out the state of the relay
  if (command.substring(3,7) == "HIGH") relayState = 1;

  else if (command.substring(3,6) == "LOW") relayState = 0;
  else return -1;

  // write to the appropriate relay
  digitalWrite(relayNumber+2, relayState);
  return 1;
}
```

http://www.instructable…

Download (https://cdn.instructables.com/ORIG/FTX/GV3Z/JBWK7NES/FTXGV3ZJBWK7NES.txt)

(https://cdn.instructables.com/ORIG/FTX/GV3Z/JBWK7NES/FTXGV3ZJBWK7NES.txt)

Step 4: Install Libraries on Your Mac

We are going to use some libraries to get the data from Zwift, analyses it, and send commands to the Photon to trigger the right relays. We need to install these libraries on our Mac.

1. Open a Terminal (cmd + spacebar and type Terminal is one way of doing this)
2. Copy paste each of the next lines into the terminal and hit enter (one by one)

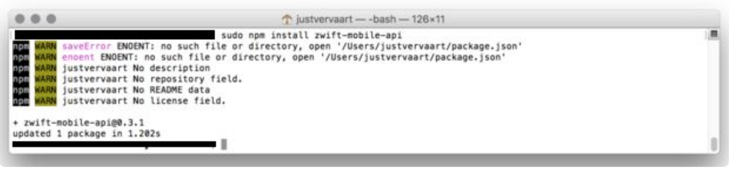
```
npm install --save zwift-mobile-api
```

```
npm install node
```

```
npm install request
```

You could see some warnings (WARN) when installing, but that should be no problem. As long as you don not see errors (ERR!). You have now installed the latest versions of the libraries needed onto your mac.

Credits: this project would not be possible without the great open source (!) Zwift API library from Ogadai



```
[redacted] sudo npm install zwift-mobile-api
npm WARN saveError ENOENT: no such file or directory, open '/Users/justvervaart/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/Users/justvervaart/package.json'
npm WARN justvervaart No description
npm WARN justvervaart No repository field.
npm WARN justvervaart No README data
npm WARN justvervaart No license field.

+ zwift-mobile-api@0.3.1
updated 1 package in 1.202s
[redacted]
```

Step 5: Edit the Javascript File

Now comes a tricky part. We need to adjust the script that get's the data from Zwift and activates the Photon to make sure it works with your credentials, both for Zwift and the Photon.

1. have your Zwift credentials (username and password) handy
2. find your Zwift ID using this online tool made by Christian Wiedmann or alternatifly via this method.
3. find your Photon Device ID & accessToken

If you have all this, download the javascript file "zwiftfan_script.js" and open it in a text editor like the free cotEditor. In the image attached you can see what lines to edit. Once you put in all the credentials, save the document with the same filename in a folder on our mac you can easily remember, like "zwiftfan"

Then some 'pro'-feature's

If you want your fan to react on different metrics like your heart rate or power output you can change

```
status.riderStatus.speed
```

into

```
status.riderStatus.heartrate
status.riderStatus.power
```

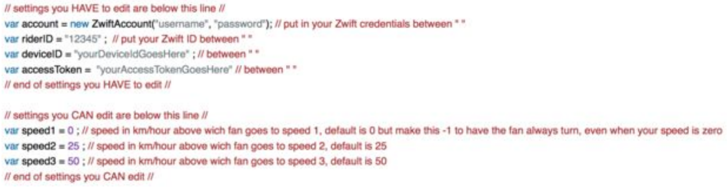
To see the metric in your terminal to see the value's that Zwift returns. You can do so by adding

```
console.log(status.riderStatus.power);
console.log(status.riderStatus.heartrate);
```

If you want to see everything Zwift return about you during a rider use

```
console.log(status); // JSON of rider status
```

*special thanks to **roekoe** for help writing and debugging the javascript code*



```
// settings you HAVE to edit are below this line //
var account = new ZwiftAccount("username", "password"); // put in your Zwift credentials between " "
var riderID = "12345" ;  // put your Zwift ID between " "
var deviceID = "yourDeviceIdGoesHere" ; // between " "
var accessToken =  "yourAccessTokenGoesHere" // between " "
// end of settings you HAVE to edit //

// settings you CAN edit are below this line //
var speed1 = 0 ; // speed in km/hour above wich fan goes to speed 1, default is 0 but make this -1 to have the fan always turn, even when your speed is zero
var speed2 = 25 ; // speed in km/hour above wich fan goes to speed 2, default is 25
var speed3 = 50 ; // speed in km/hour above wich fan goes to speed 3, default is 50
// end of settings you CAN edit //
```

Download (https://cdn.instructables.com/ORIG/FN3/E2L4/JC0U6NCU/FN3E2L4JC0U6NCU.js)

http://www.instructable…

(https://cdn.instructables.com/ORIG/FN3/E2L4/JC0U6NCU/FN3E2L4JC0U6NCU.js)

Step 6: A Launch Script

You can activate the program by navigating in your Terminal to the folder where you saved it and then type

```
node zwiftfan_script.js
```

and press enter.

But that's not very conveniënt if you are on your bike and forgot to launch the program. And also, sometimes the program will crash (don't no why, if anybody does, please let me know) and the script below will make the program relaunch automagicly. So i made a shell script you can double click. Here we go: ->

In your text editor create a new file and paste this code into it:

```
#!/bin/bash
cd "$(dirname "$0")"
until node zwiftfan_script.js; do
    echo "Zwiftfan crashed! ...respawning..." >&2
    sleep 1
done
```

Save the file as "*start_zwiftfan.sh"* in the same folder as your "zwiftfan_script.js" file is.

Rightclick on the file in your Finder to get options. Select '*Open with*' and '*other*'.

At the bottom of the next screen check the box with '*Always open with this program'* and choose '*All programs'* in the dropdown just above that check-box. Then click the '*Open' button.*

One more thing tricky thing;

- Open your Terminal (CMD + spacebar and type Terminal + ENTER)
- type;

```
cd [name of your directory]
```

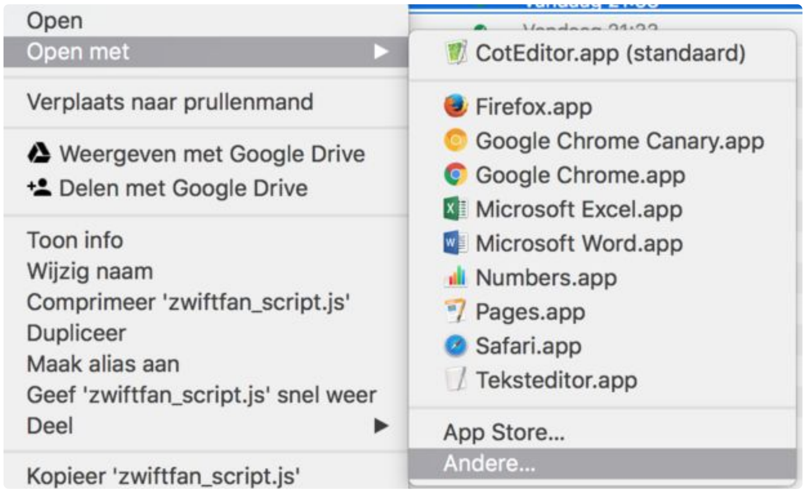
press enter and then type

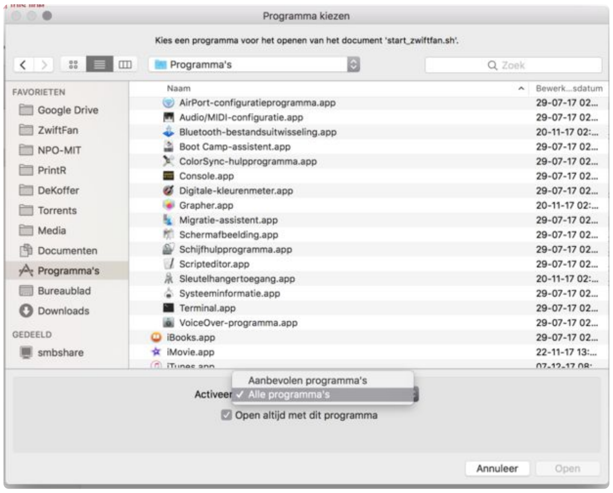
```
chmod 700 start_zwiftfan.sh
```

and enter again.

You now have made a file you can double-click to start our javascript program with the Terminal. Or launch with one click if you put it in your launchbar. If Zwift is running the Terminal will print the current speed in Zwift every second. If Zwift is not active the script will return errors.

*ps. sorry, the screenshots are in Dutch, but i think you will manage. ;) Otherwise you just have to learn Dutch, but don't worry, Dutch is easy! Just say "*stroopwafels*" and smile.*





Kies een programma voor het openen van het document 'start_zwiftfan.sh'.

Naam	Bewerk...sdatum
AirPort-configuratieprogramma.app	29-07-17 02...
Audio/MIDI-configuratie.app	29-07-17 02...
Bluetooth-bestandsuitwisseling.app	20-11-17 02:...
Boot Camp-assistent.app	29-07-17 02...
ColorSync-hulpprogramma.app	29-07-17 02...
Console.app	29-07-17 02...
Digitale-kleurenmeter.app	29-07-17 02...
Grapher.app	20-11-17 02:...
Migratie-assistent.app	29-07-17 02...
Schermafbeelding.app	29-07-17 02...
Schijfhulpprogramma.app	29-07-17 02...
Scripteditor.app	29-07-17 02...
Sleutelhangertoegang.app	20-11-17 02:...
Systeeminformatie.app	29-07-17 02...
Terminal.app	29-07-17 02...
VoiceOver-programma.app	29-07-17 02...
iBooks.app	29-07-17 02...
iMovie.app	22-11-17 13:...
iTunes.app	07-12-17 08:...

Aanbevolen programma's
Activeer ✓ Alle programma's

☑ Open altijd met dit programma

Step 7: Some Final Words

I hope it all worked in the end. If you have used this tutorial I would love to hear from you and maybe a picture? And if you have any improvements to the project or these instructions, feel free to send me an e-mail at just@justvervaart.nl

Happy Zwifting!