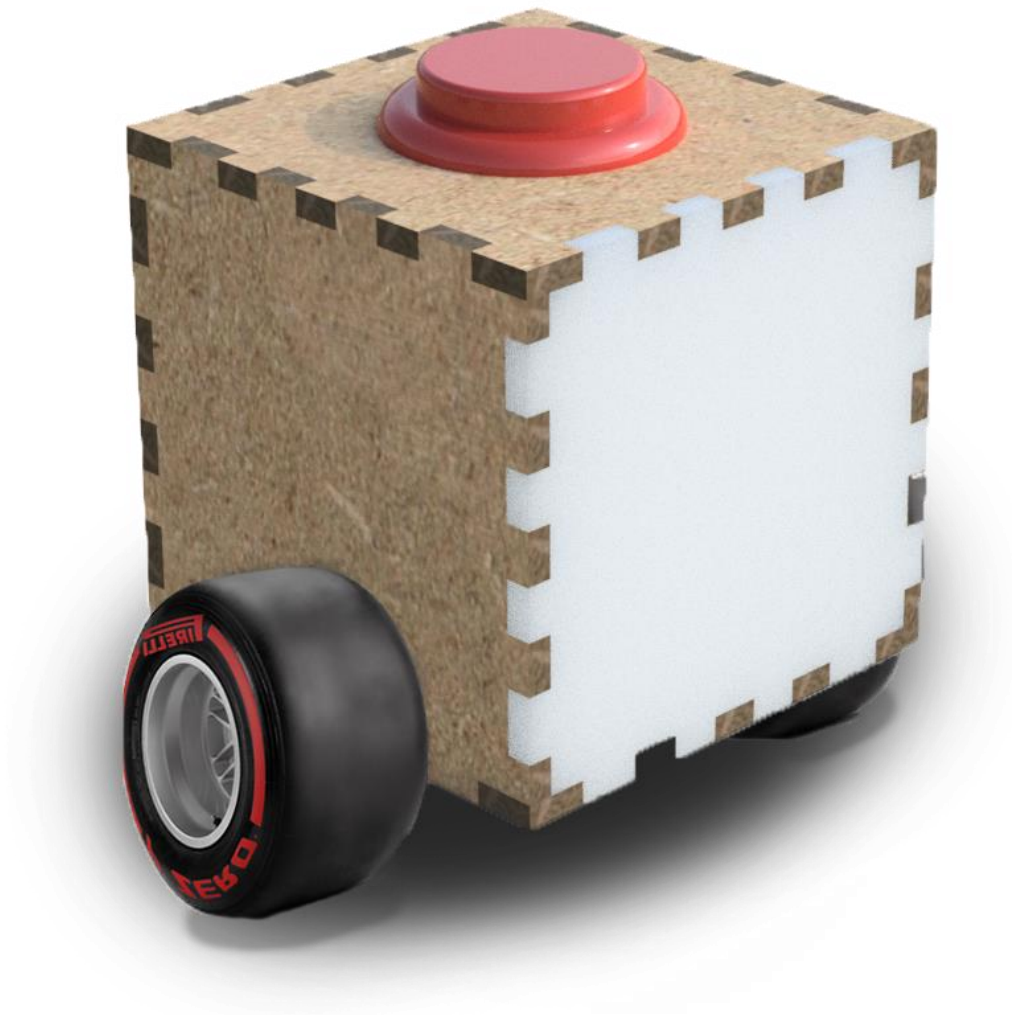


# BoxBot Documentation

Arnout Schekkerman IT301 IoT Workshop 2016-2017



## Table of content

The Idea.....	3
The logo language .....	3
Web interface .....	4
Hardware .....	7
Arduino code.....	8
Expansion/problems I faced .....	9
Sources.....	9

## The Idea

The idea for this project was to make a the box drivable using a web interface. The idea grew to a Seymore Papert like logo turtle<sup>1</sup>. Seymore Papert was a mathematician, computer scientist, and educator, who spent most of his career teaching and researching at MIT. He was one of the pioneers of artificial intelligence, and of the constructionist movement in education. He was co-inventor, with Wally Feurzeig, of the Logo programming language. This programming language was designed to teach children about math using a robot that drew a line on paper and that could be controlled using a computer with the logo programming language. This project is going to be used to teach children about programming in a fun and hands-on approach, just like Seymoure Papert did with his turtle.

## The logo language

I based the grammar on the logo language that Seymour Papert used because this project and its target group is almost completely the same as Paperts project and target group. Papert designed the code to educate children about math using a robot and a language to control the robot. My goal is to educate children about programming using a robot. The way they would learn about programming is hands-on and visible in the real world.<sup>2</sup> This way the errors you make will also become apparent in the real world.

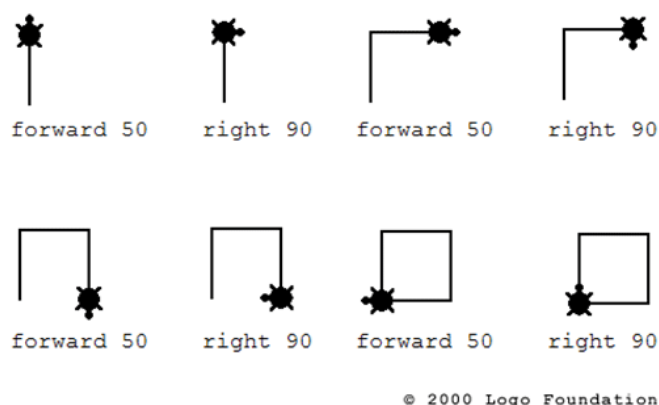


Figure 1 example of the logo language

I did not have a lot of time so I implemented a fraction of the language, for more information about the full language follow the link<sup>3</sup> below.

I designed two ways to control the BoxBot: chain modus and code modus. Chain modus is like using code blocks and code modus is just an input field for your code. Follow the link<sup>4</sup> for a live example

<sup>1</sup> [http://el.media.mit.edu/logo-foundation/what\\_is\\_logo/index.html](http://el.media.mit.edu/logo-foundation/what_is_logo/index.html)

<sup>2</sup> [http://el.media.mit.edu/logo-foundation/what\\_is\\_logo/logo\\_and\\_learning.html](http://el.media.mit.edu/logo-foundation/what_is_logo/logo_and_learning.html) paragraph 4

<sup>3</sup> [http://el.media.mit.edu/logo-foundation/what\\_is\\_logo/logo\\_programming.html](http://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html)

<sup>4</sup> <http://arnoutschekkerman.me/boxbot.html>

## Web interface

The web interface exist of 3 pages: the index, chain modus and the code modus.

Below is a diagram that shows how the web interface goes from code to command for the API.

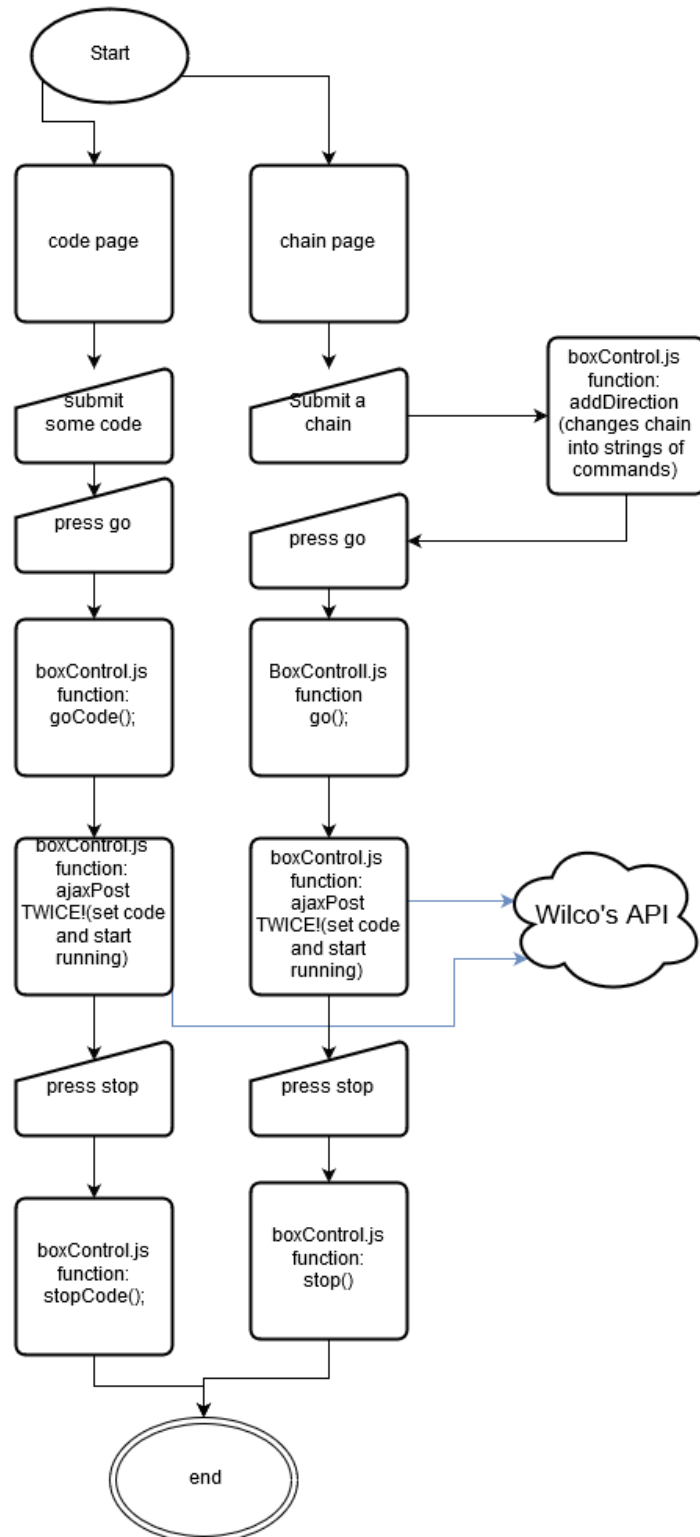


Figure 1 code to API command

### Chain modus:

When a user adds a command in chain modus the addDirection function of boxControll.js adds an image to the page and adds a command + speed + time (including dividers) to the directionArray. When you press go, all the array entries get concatenated and get sent to the nodeMCU using the API.

When a user removes a command the function erase gets called. The last command gets removed and the last array entry gets also removed.

The stop button send a predetermined string("stop") to the BoxBot.

### Code modus:

Code modus is like chain modus only it does not make use of addDirection because the user makes the string themselves.

I created separate go() and stop() functions(goCode() and stopCode()) because the code has to be handled differently, the code has to be retrieved from the text space and spaces have to be removed.

I also create a different stop function as a quick fix, the normal stop would change the goCode() button to a normal go() button after clicking on the stopCode button.

### Communication overview

The diagram below is the complete overview of how the box is controlled.

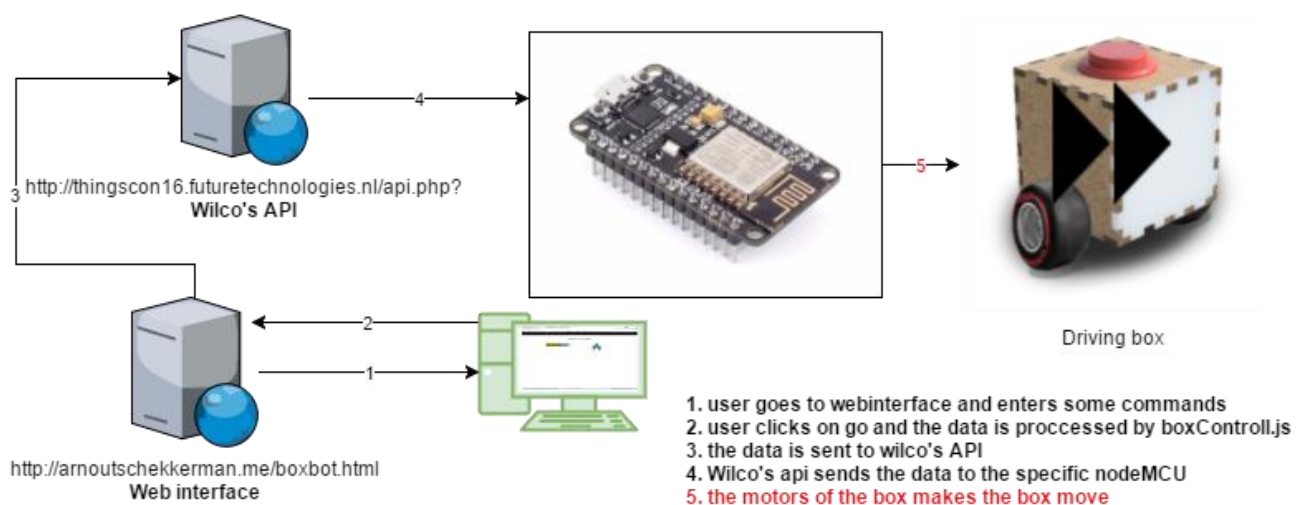


Figure 2 how a command makes the box drive

Below is the API help page from Wilco's website, it explains the way the URL I use for posts are formulated:

I use the following URLs:

"http://thingscon16.futuretechnologies.nl/api.php?t=sd&d=FF6E&td=FF6E&c=&m=" + recordedDirections; and "http://thingscon16.futuretechnologies.nl/api.php?t=sqi&d=FF6E";

These set the command(message on the API) and send the message to the NodeMCU. You can see that in the code below.

```
//post recordedDirections to wilco's api and then to the nodeMCU, FIRST! Set
device configuration (api url with t=sd)
urlSDC =
"http://thingscon16.futuretechnologies.nl/api.php?t=sd&d=FF6E&td=FF6E&c=&m=" +
recordedDirections;
successSDC = console.log("go SDC post success");
ajaxPost(urlSDC, successSDC); //ajaxPost, see ajaxPost function

//SECONDLY the Set query item (api url with t=sqi)
urlSQI = "http://thingscon16.futuretechnologies.nl/api.php?t=sqi&d=FF6E";
successSQI = console.log("go SQI post success");
ajaxPost(urlSQI, successSQI); //ajaxPost, see ajaxPost function
```

Table 1 Wilco's API help

You can use the following parameters in a querystring:

Parameter	Datatype	Description
t	string	Type of action
d	string	Device id
td	string	Target device id
c	string	Hex color
v	int	Version

Type of actions (t):

\* is required

Key	Description	Parameters	Output
sd	Set device configuration	d*, td*, c*	1 or -1
rd	Remove device configuration	d*, td*	1 or -1
gq	Get query item	d*, v	hex_color(,spring_constant,damp_constant,message)
sq	Set query item	d*	1 or -1

#### Examples:

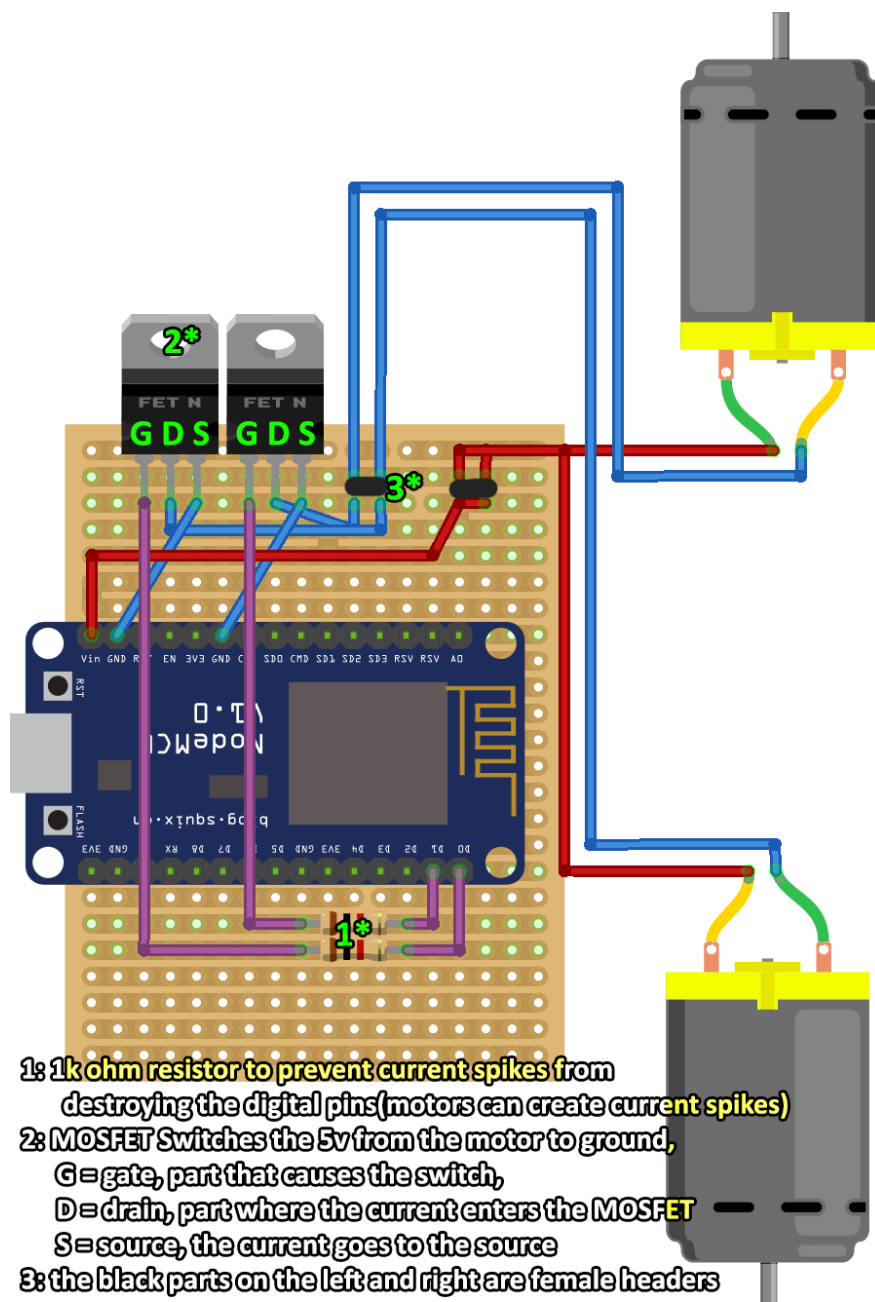
api.php?t=sd&d=T111&td=T222&c=ff0000	Sets the device configuration of T111 for T222 with the hex color ff0000
api.php?t=gq&d=T111&v=2	Gets the next queue item for T111, v=2 means that it will not only return the color but also the spring constant, damp constant and message of the queue item

## Hardware

The hardware consists of two motors, two MOSFETs, two 1k ohm resistors, and a NodeMCU. The motors are 5v and driven by the 5v Vin/VV pin. The 3v D1 and D0 pins can't drive a motor because the voltage and current is too low. The D1 and D0 pins are used to switch the motors on and off as well as using PWM to control the speed of the motor. The on and off switching is done by the MOSFETs. A MOSFET has 3 pins: a gate(that is where the D1/D0 pin is connected), a drain(connected to a motor) and a source pin(connected to the ground).

When D1/D0 is high the gate switches the MOSFET so current can flow, the motor is already connected to 5v. when the MOSFET switches the current will flow through the MOSFET to the ground pin thus making a loop thus driving the motor.

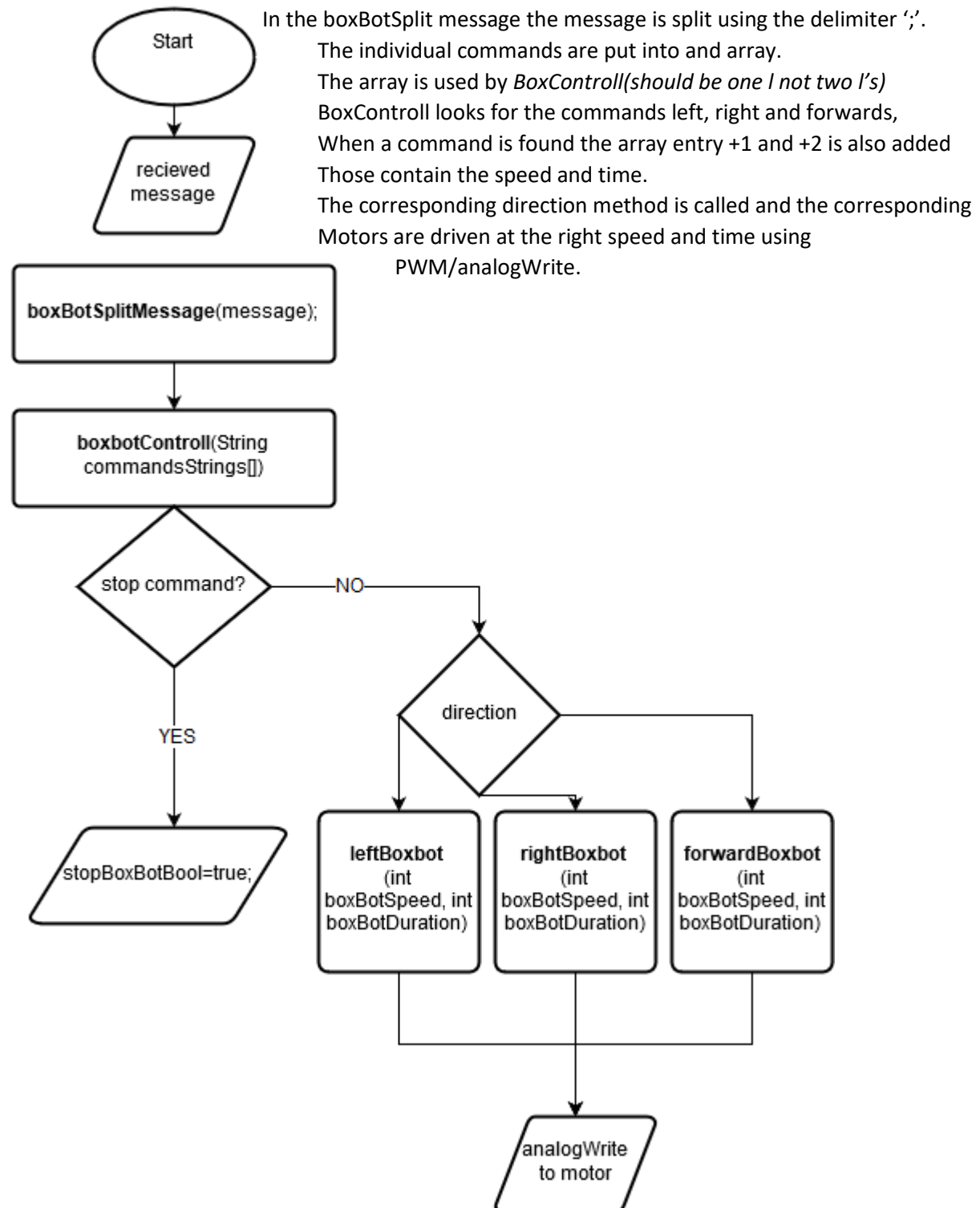
The motor can keep turning for a while after the power has been cut. it is able to work as a generator, this can create a current, that is why I've protected the D1/D0 pin with a 1k ohm resistor. Although it is unlikely I added the resistors just to be safe.



Blue wire = ground, Red wire = 5v, purple wire = 3v/data fritzing

## Arduino code

The code in the Arduino receives messages every 2 seconds. When the message with the commands is sent the following happens(as described in the flowchart below).





## Expansion/problems I faced

Expansion:

- Expand the language to the full logo language.
- Make the website more responsive for mobile.
- Allow the users to save the code in browser.
- Allow one stopbutton to control the code and the chain modus
- Code highlighting.
- Allow the motors to reverse polarity so the box can move backwards.
- Create your own API because Wilco will take his Server down sometime in the future.
- Allow the Stop command to interfere with running commands by using threads or some kind of interrupt.

The problems

- The Arduino code crashes sometimes.
- The MOSFETs won't switch correctly.

## Sources

**API and communication code of the box(the project BoxBot was built upon):**

[https://github.com/Exxon1800/IOT\\_Workshop](https://github.com/Exxon1800/IOT_Workshop)

**the box website (the project BoxBot was built upon):**

<http://thingscon16.futuretechnologies.nl/>

and [http://thingscon16.futuretechnologies.nl/api\\_help.php](http://thingscon16.futuretechnologies.nl/api_help.php)

**The BoxBot webinterface:**

<http://arnoutschekkerman.me/boxbot.html>