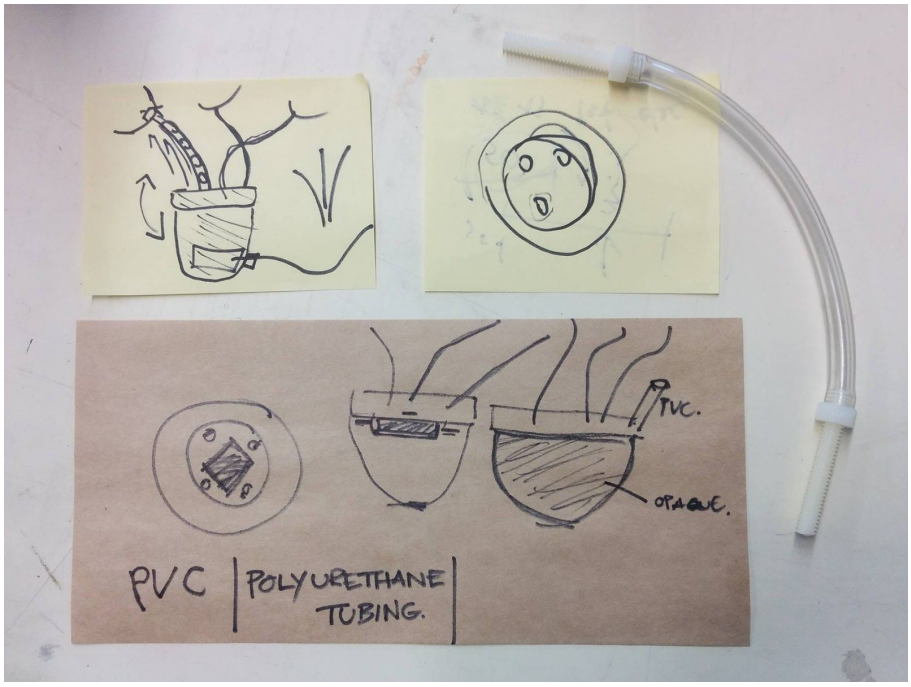
Vsevolod Ivanov
Concordia University
seva@tumahn.net

HackerPot

Fall 2016

Process

Concept

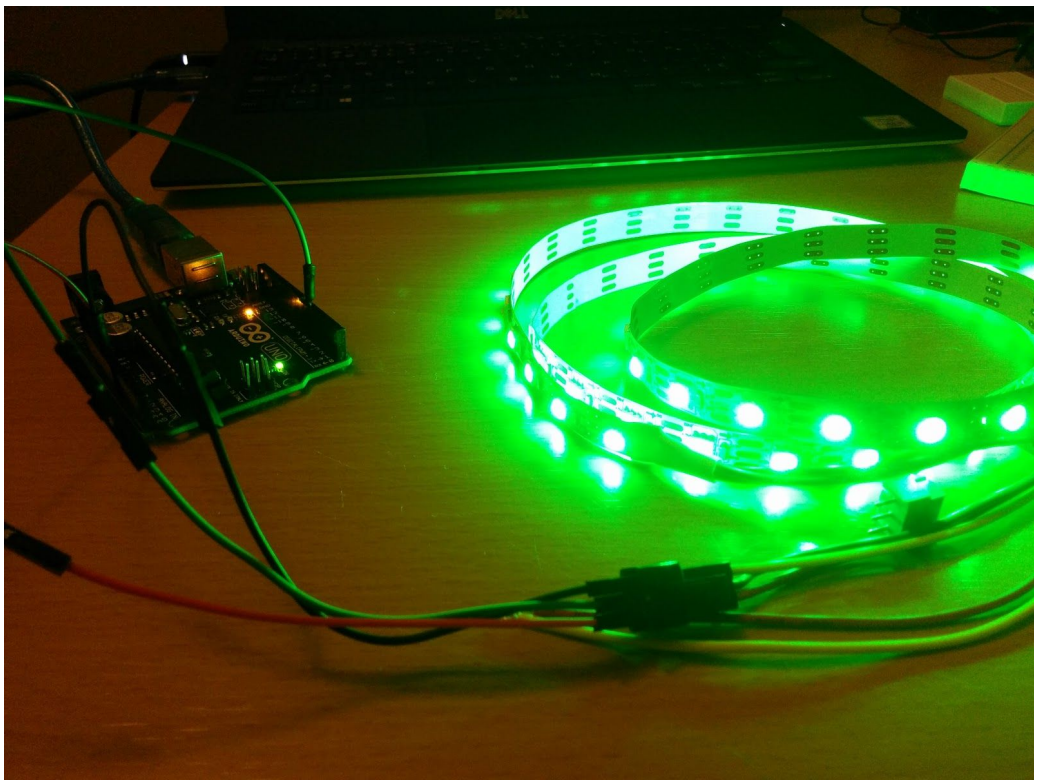


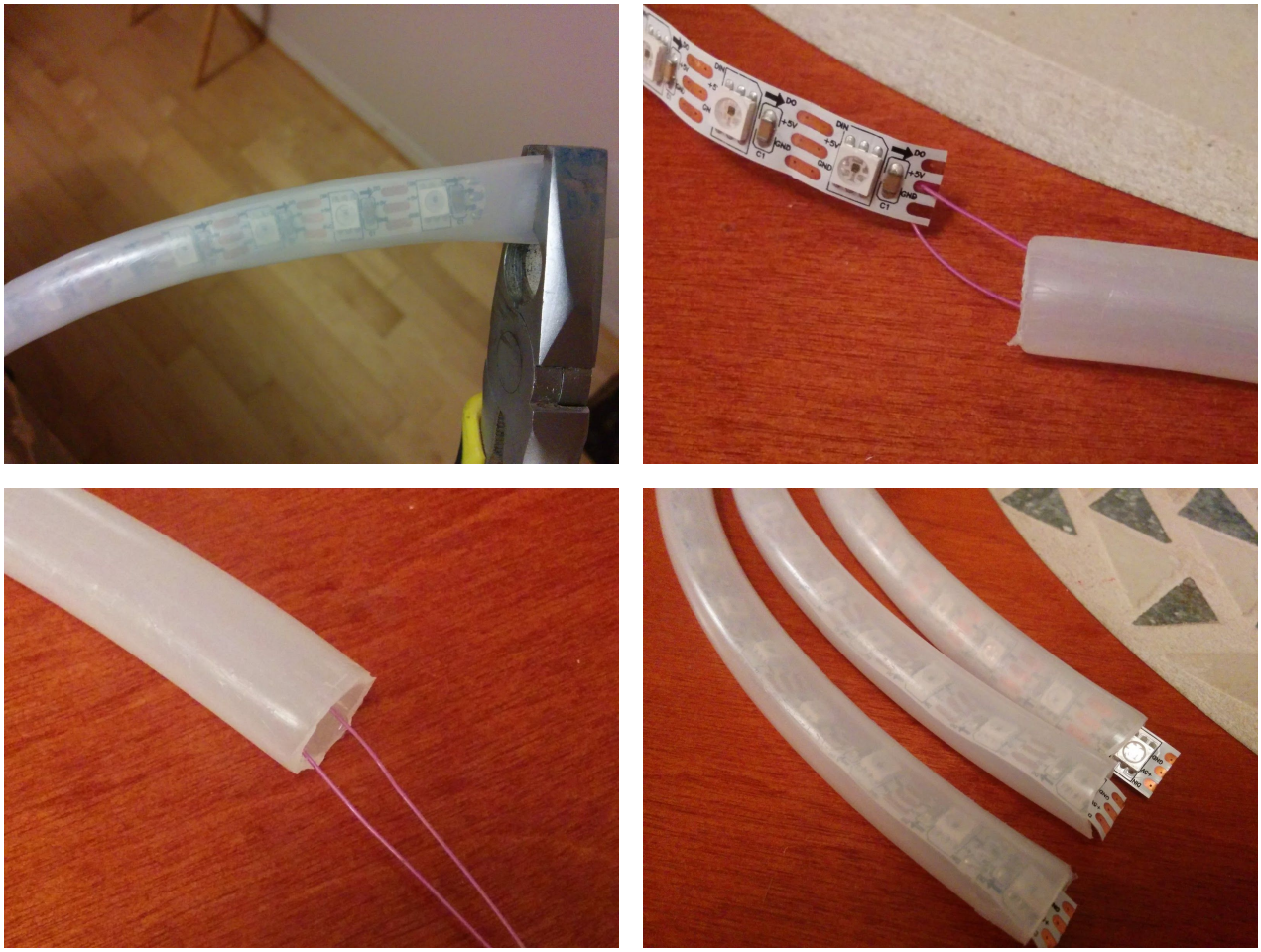
Translucid Pot

After long research of possible translucid spheres in multiple stores, I came to realize that it will have to be a lamp component :

Finally, I found these which suited my needs :

Stems





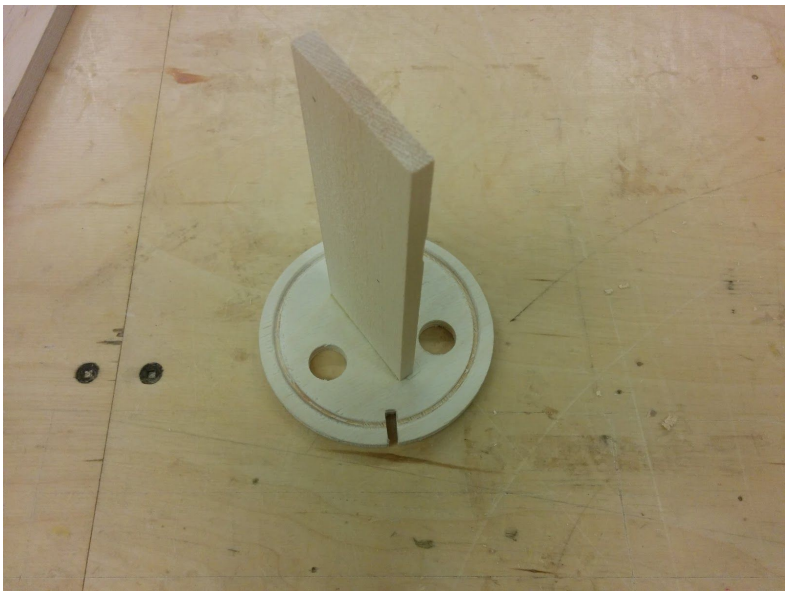
That was a long and tedious task.

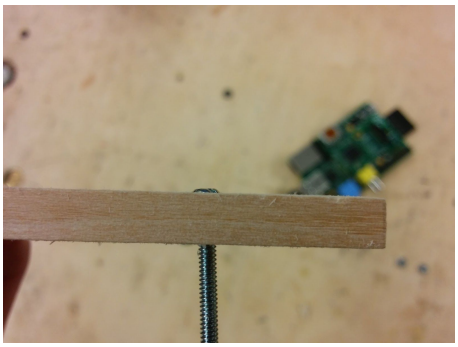
Pot + Stems

It was hard to tell but I prefered the translucid ones due to their uniformity :

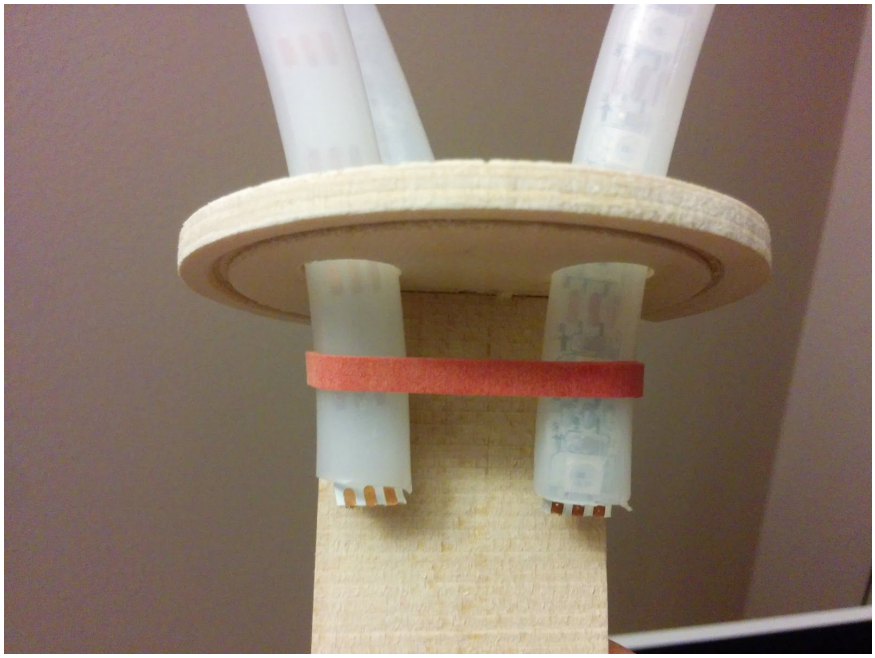
Wood structure

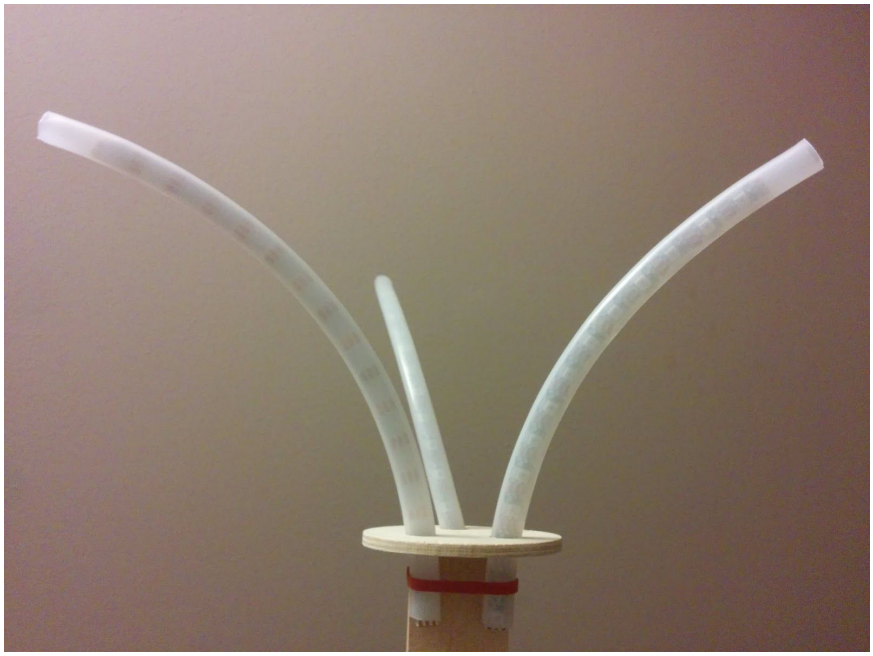
I settled for Concordia's Woodshop adventures to build a custom support for the hardware:



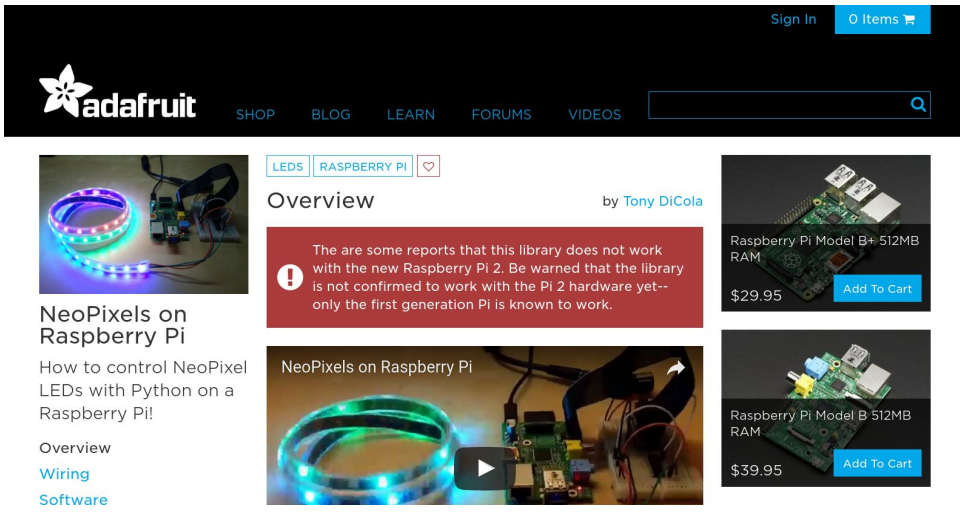


Stems + Wood Structure

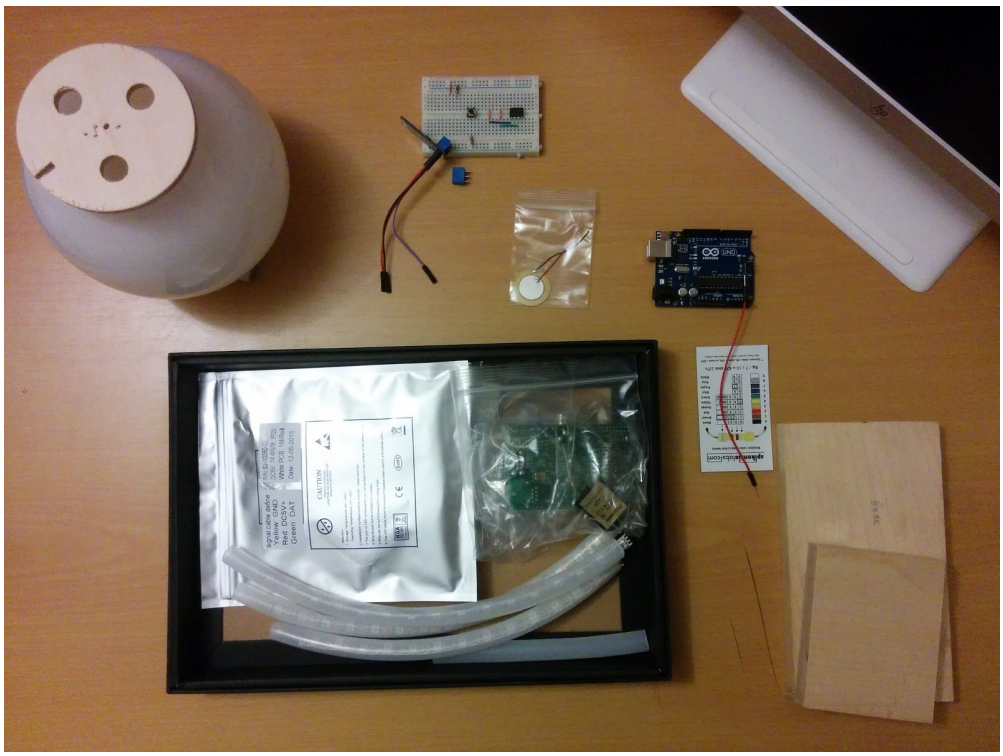




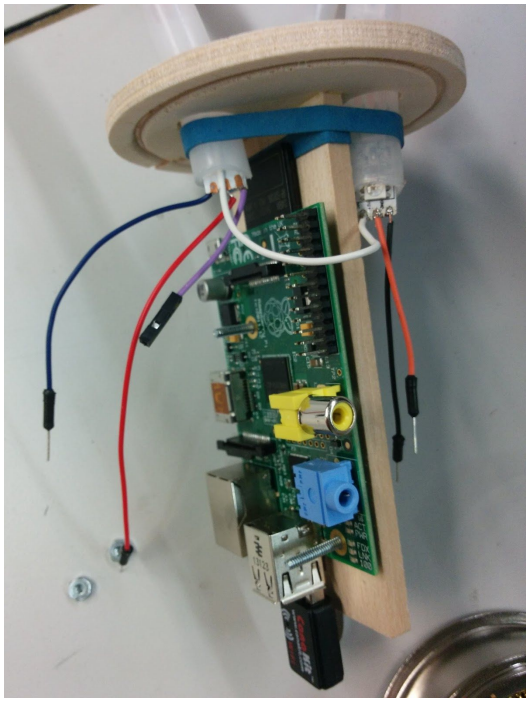
Hardware

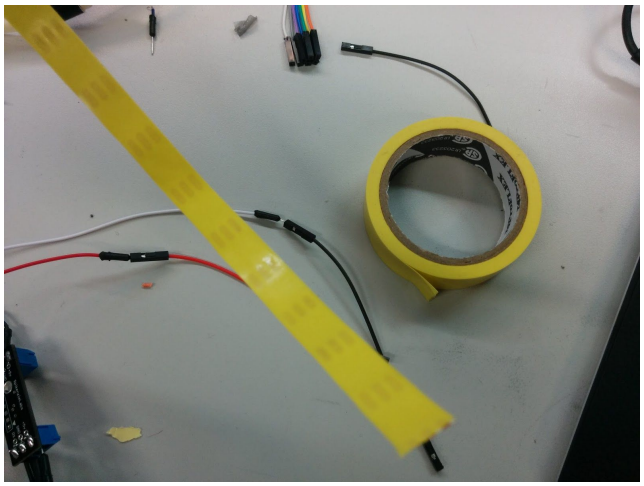


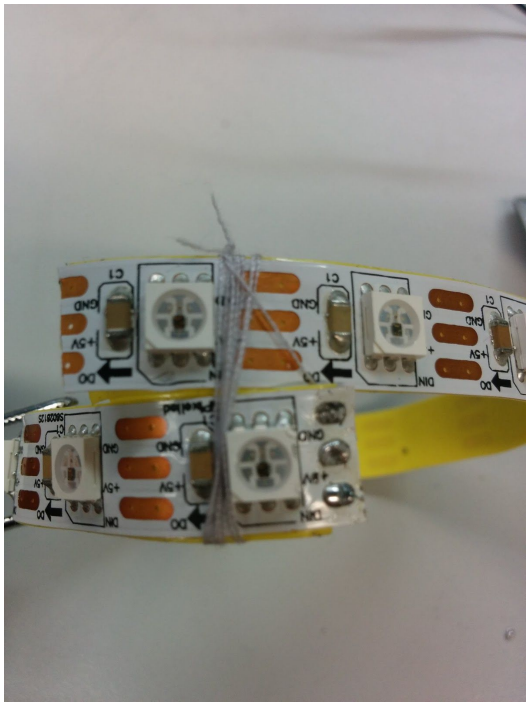
I used a Raspberry Pi due to intense network tasks. This Neowpixel library wasn't precise enough so I used SPLixel from Spekenzielabs to address all of the Neopixel strip in UART serial



:

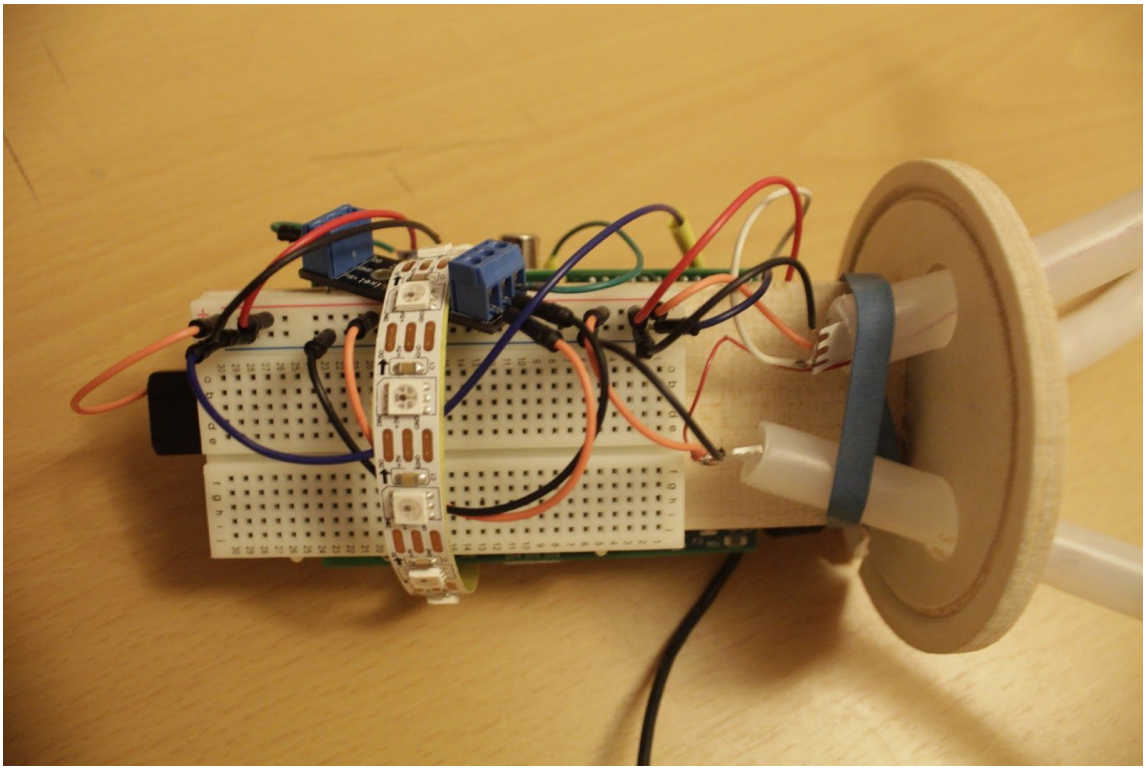


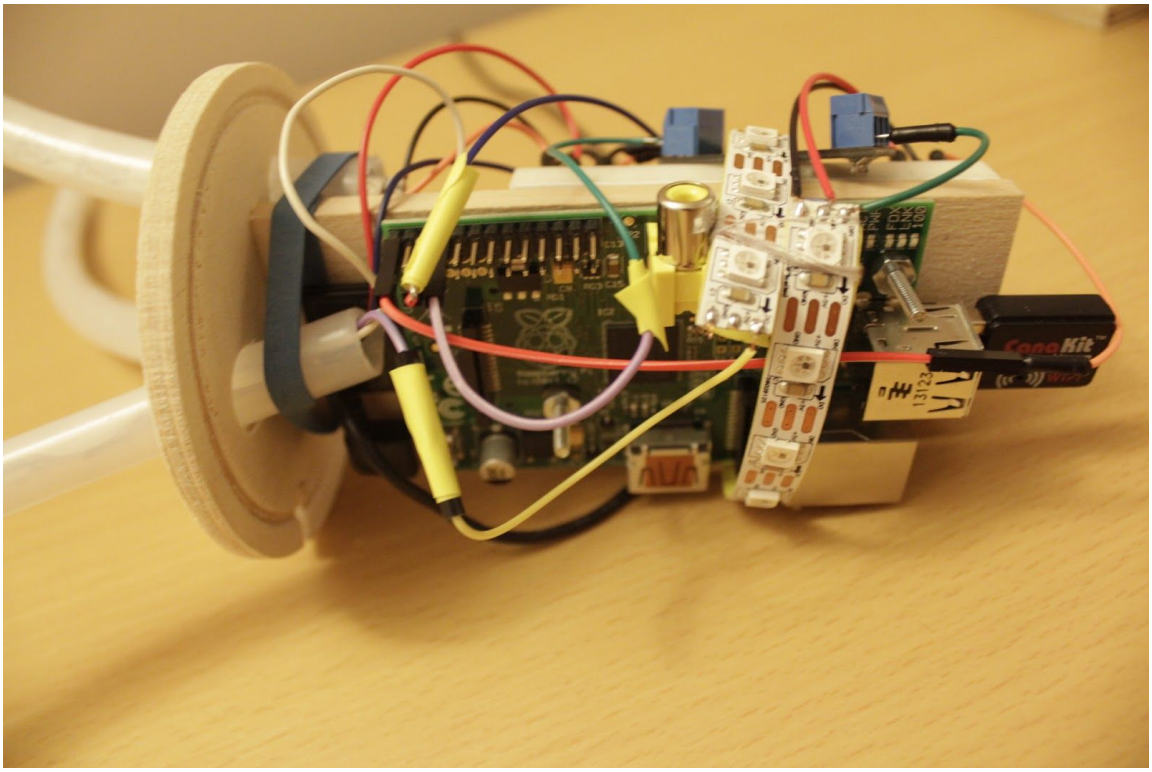




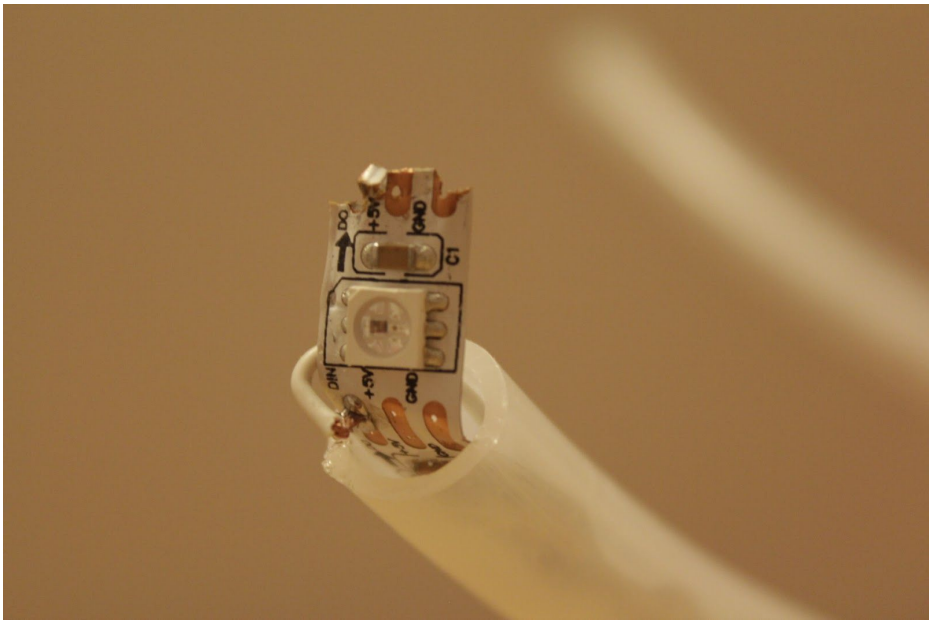
All of the Neopixel LED strips have DIN (data in), DOUT, 5V and GND lines. The goal is to power all of the LED strips from Raspberry Pi 5V and then interconnect their DIN to DOUT between the strips to address them individually using the serial commands. The SPLixel is a microcontroller allowing you to remove all Raspberry Pi interference due to internal tasks and to address each LED individually by sending 4 bytes according to its datasheet.

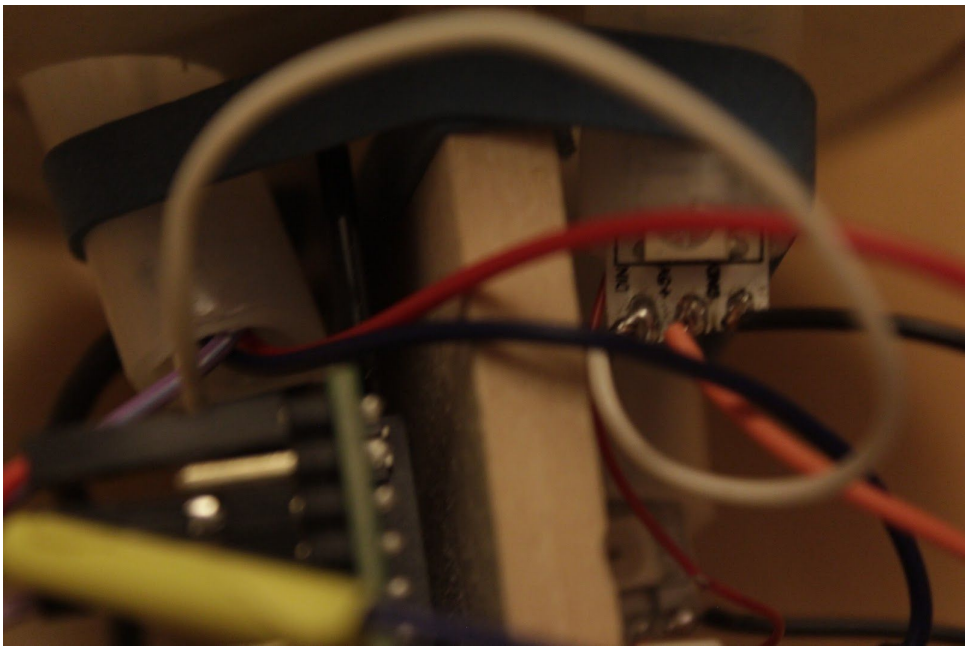
Here is the circuit layout of the HackerPot :



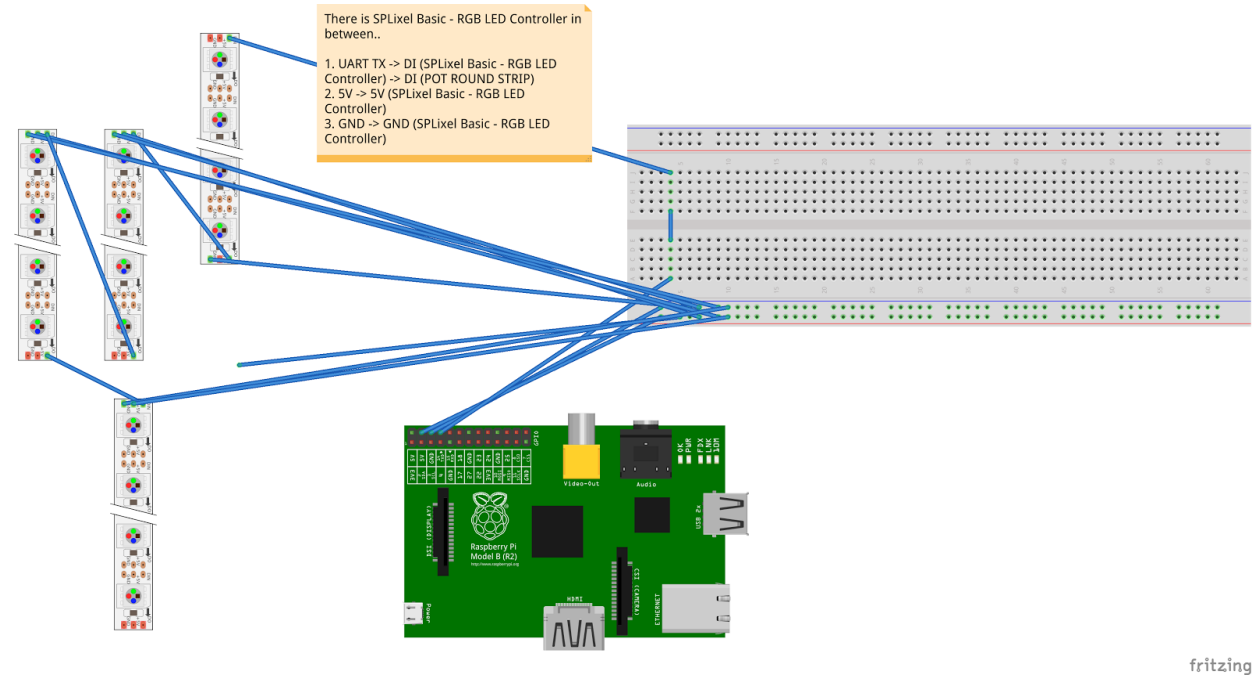


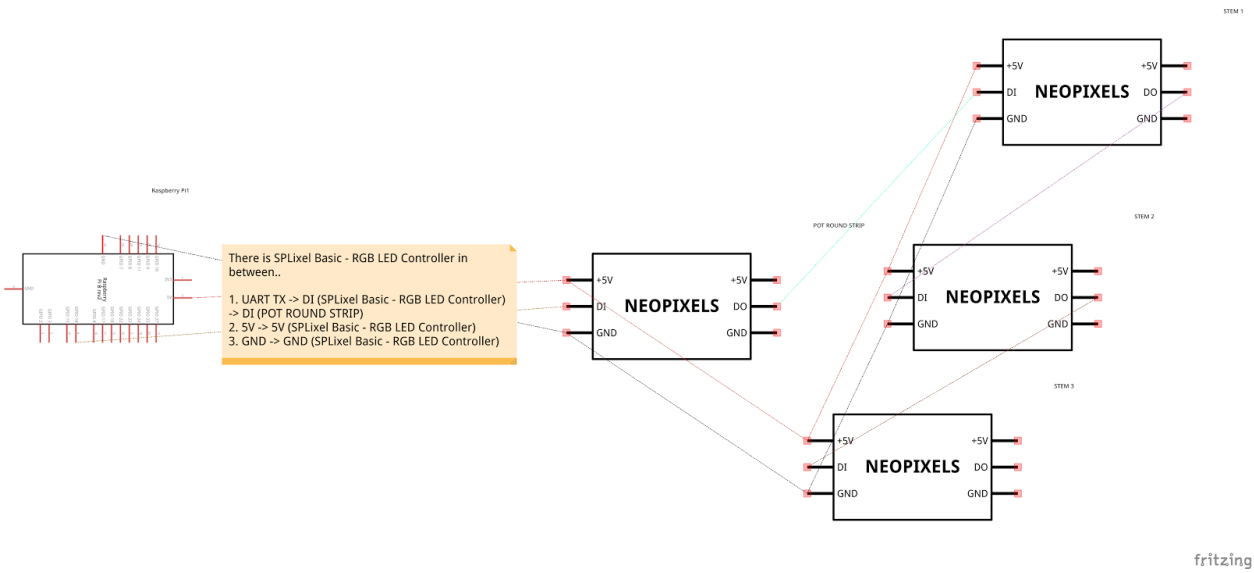
For DOUT (data out) I dropped a wired from the top of the step to the DIN at the bottom :





Circuit

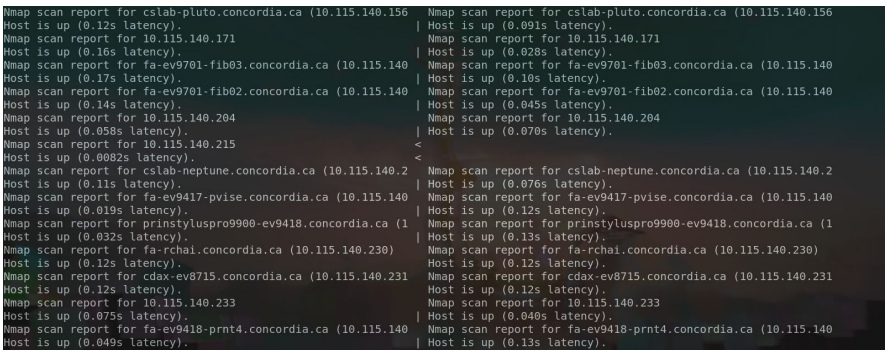

There is SPLixel Basic - RGB LED Controller in between..
1. UART TX -> DI (SPLixel Basic - RGB LED Controller) -> DI (POT ROUND STRIP)
2. 5V -> 5V (SPLixel Basic - RGB LED Controller)
3. GND -> GND (SPLixel Basic - RGB LED Controller)
Raspberry Pi Model B (R2)
fritzing


There is SPLixel Basic - RGB LED Controller in between..
1. UART TX -> DI (SPLixel Basic - RGB LED Controller) -> DI (POT ROUND STRIP)
2. 5V -> 5V (SPLixel Basic - RGB LED Controller)
3. GND -> GND (SPLixel Basic - RGB LED Controller)
+5V
DI
GND
NEOPIXELS
DO
STEM 1
STEM 2
STEM 3
POT ROUND STRIP
fritzing

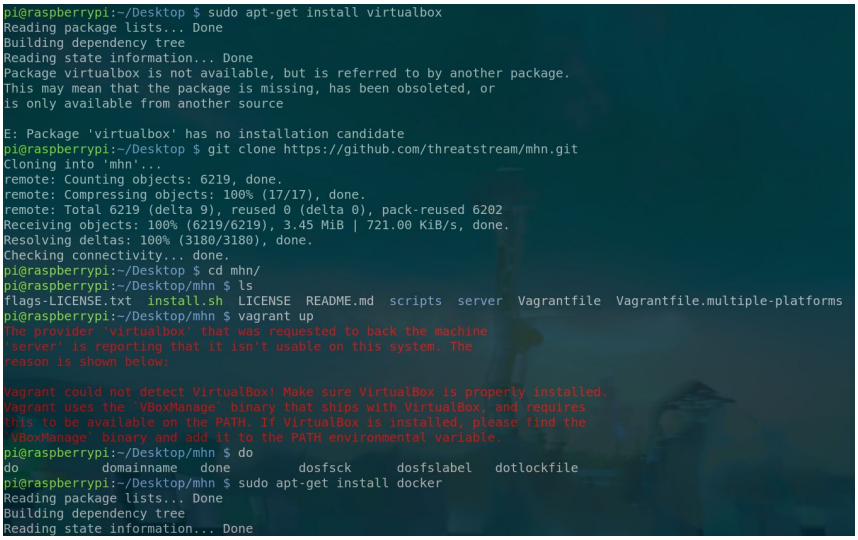
Virtual

I had some troubles at finding my device in University. When I connected it to the network. I had no idea of its IP address and scanning the network was blocking me after a while. Plus, when I did a loop on all subnet to SSH on every IP, I did hit myself HoneyPots which were keeping the connection alive without any further response...

I found a way to find it by using one scan results of “nmap -sn <ips>” with the Raspberry Pi connected to the network and one another scan after disconnecting it. Then, I did an “sdiff” on both results and looked for a difference :

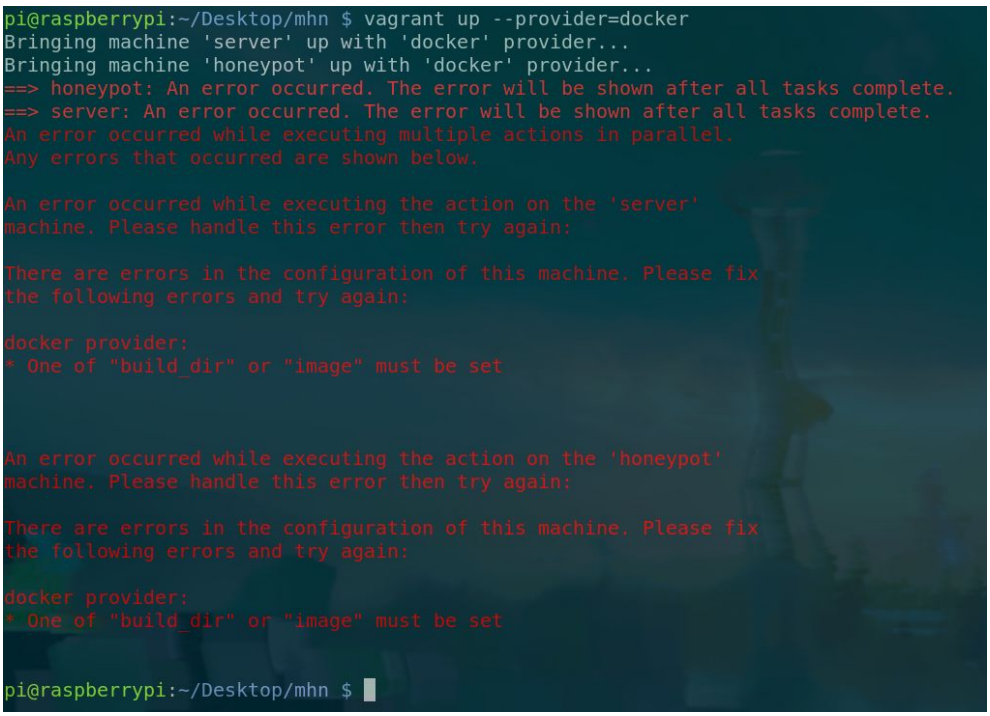


At first, I thought of going with MHN (Modern Honeypot Network) to create this Honeypot for trapping the targets. However, turns out you need to virtualize them to allow a more high level interaction. I decided to install virtualbox as recommended :



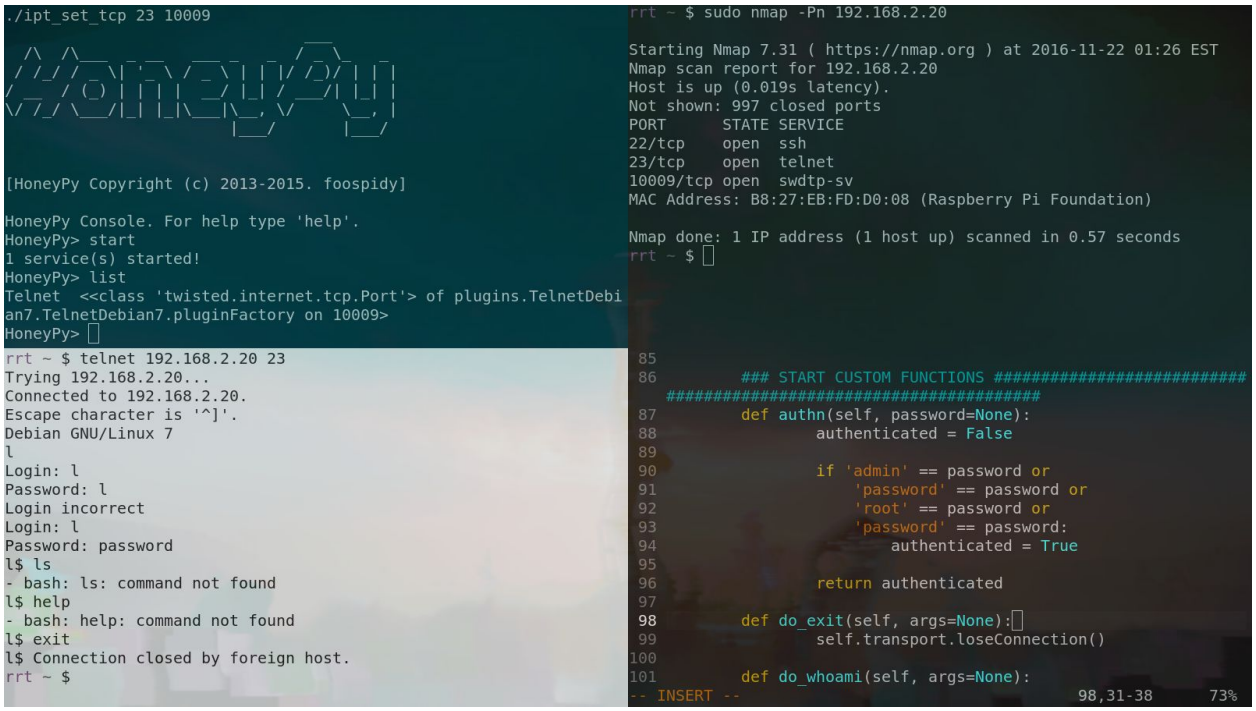
To discover that you can’t due to Raspberry Pi limited capabilities (at least not on Raspbian).

Then, I started exploring docker but it needs a config file :



I found one in their repositories but this all brought me to search for something more customizable since I needed a very precise triggering of the attacks...

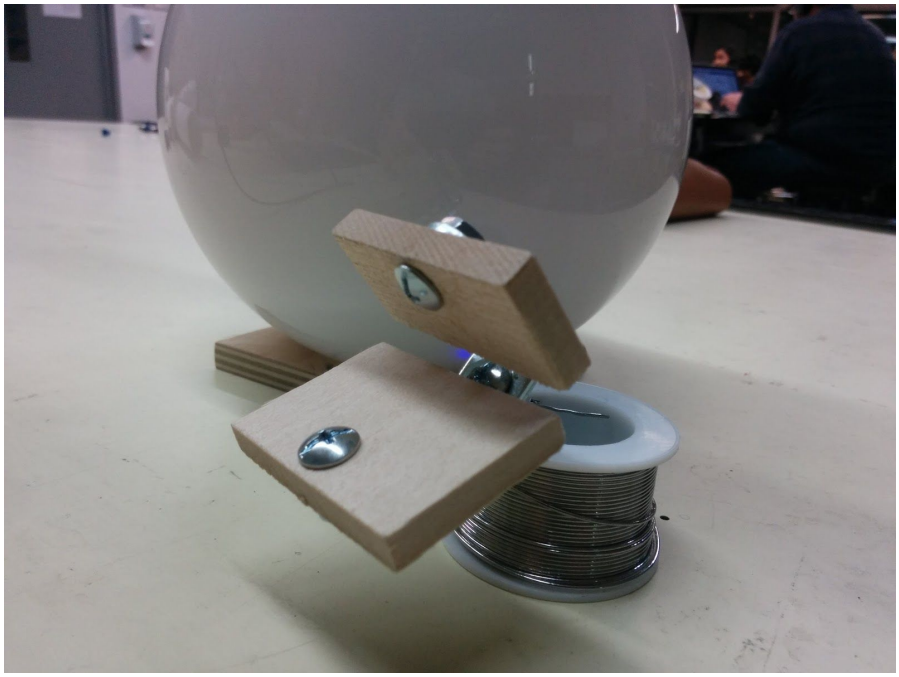
I saw previously a library in Python called Twisted that allowed to directly extend TCP / UDP to create the desired protocols. This lead me to discover HoneyPy which is build on top of it. I used it with a combination of its Telnet plugin that I modified:



Flowers

I started by looking in the hardware store for Venus Flytrap alike shapes :

I tried to build some myself but I wasn't satisfied with the result :



Then, I found these which fitted the structure perfectly :

From there I put everything together as seen in prototype videos!