**Computer/Network Side:**

TODO: none really.

Moonshot: add cpu temperature monitoring to app???

---

7/26/17: All of the method of sleep/shutdown and restart failed for one reason or another; I just got a fan and keep the cpu constantly running and cool. Will probably manually shutdown occasionally.

Because we can’t have happy/proper IoT thanks to PittNet, we’re using a pc that forwards PittNet and acts as a hotspot:

SSID: B09IoTNet

Pass: Fireproof12

----

6/14/17

To prevent the computer from overheating, along with*BUYING A FAN (done)***,** the cpu should go to sleep for 5hrs every day (or so)

For network setup/teardown scripts:

https://www.windowscentral.com/how-turn-your-windows-10-pc-wireless-hotspot

Commands needed after initial setup are:

startup:

NETSH WLAN set hostednetwork mode=allow ssid=B09IoTNet key=Fireproof12

NETSH WLAN start hostednetwork

Stop (and sleep):

NETSH WLAN stop hostednetwork

Rundll32.exe Powrprof.dll,SetSuspendState Sleep

Use Task Scheduler for managing running scripts

--

FOR SOME REASON THE SLEEP WAKEUP DON’T WORK

So instead, let’s do this: <https://www.nextofwindows.com/how-to-schedule-windows-automatically-power-on-and-power-off>

UGH this also failed bc we don’t have power options in our BIOS for some reason… att3:

<http://www.hp.com/hpinfo/newsroom/press_kits/2010/plugandprint/pdf/HPAOAO_Solution_Brief.pdf>

Nvm that’s just for printers

*So:*

1. *In Task Scheduler, create task for executing script to close network at time X*
2. *In Task Scheduler, create task for Shutdown at time X+5min*
3. *In Task Scheduler, create task for executing launching network script at time X+5h15*
4. *In BIOS, set automatic startup at time X+5 hours or so*

That’s everything on this end, we can add intermittent quick shutoffs and stuff during the day if it’s helpful to keep the cpu cool.

---

All failed…. See above.

**ESP8266/Blynk Side:**

TODO:

Current scripts should connect to the wifi network if it’s there on boot. If it’s not is the issue. So:

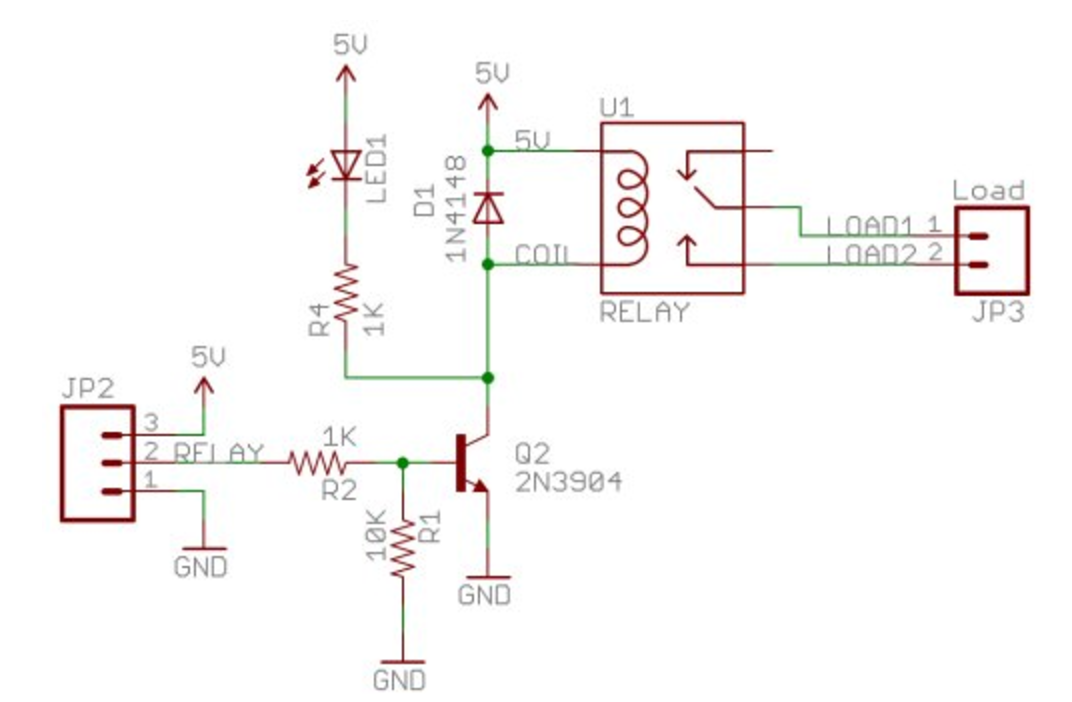
1. *Ensure that current script for it works as planned (connects if B09IoTNet is present)*
2. *Adjust script to reconnect every X minutes if connection is lost*
   1. *This is not only important for connection drops, but also obviously for when the CPU that’s acting as a router shuts off for 5ish hours.*
   2. *WCS resetting should work, as that normally forces it to reconnect. If software reset it possible, that’d work, or, even worse, if needed we can power a servo to manually hit the reset line… ew.*
3. *Temperature checking of CPU? Would be nice but possibly tricky*

**Turning On Coffee Pot Side:**

TODO:

* *Ensure all stuff works w/ the relay and doublecheck the setup is good*
* *Make PCB with all this stuff and setup (and enclosure? That’d be cool)*

Most of this should be functioning, as the circuit is there, may need to be reset up though. And enclosure for the outlet would be good...For reference: <https://www.sparkfun.com/tutorials/119>



There are a few cool things we could do with this though!

**Water Distribution Side:**

Most of this should be in the clear - the scripts should have the calculated time amounts needed for distribution. One thing to think about could be alerting if water amount is low….how could we do that?

TODO:

* *Ensure all stuff in action currently works still*
* *Consider trying to figure out water level problem?*

**Coffee Distribution Side:**

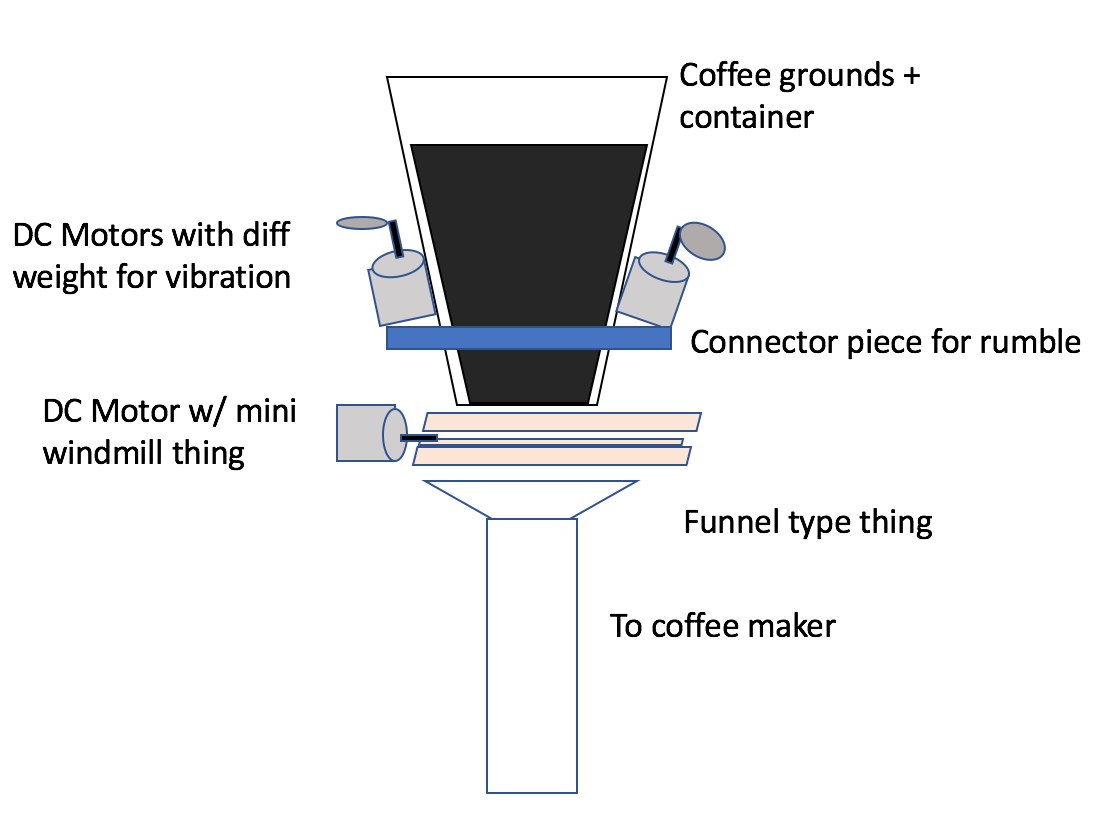
TODO:

1. *Construct prototype to see if this works*
   1. *Take cardboard coffee cup for reservoir, construct setup, get crappy coffee to use*
   2. *See how fast dc motor should spin, make sure enough power to move coffee, optimize “blade” size*
   3. *Send a command to spin X amount per cup of coffee, research equation for that*
2. *Construct real thing*
   1. *Material for reservoir... Could use coffee cans -> “funnel (pretty wide) -> cardboard cylinders? Plastic would be better, maybe old 2L soda bottles?*
   2. *Drill big hole in top of coffee maker and fix the setup. Need to figure out size of entry before doing. Remember that we want the coffee to come in at an angle.*
   3. *Implement script with calculation for calculating spins/cup of coffee*

---

7/10/17:

Previous idea ditched bc coffee gets stuck; flicks are inconsistent. New new idea is:

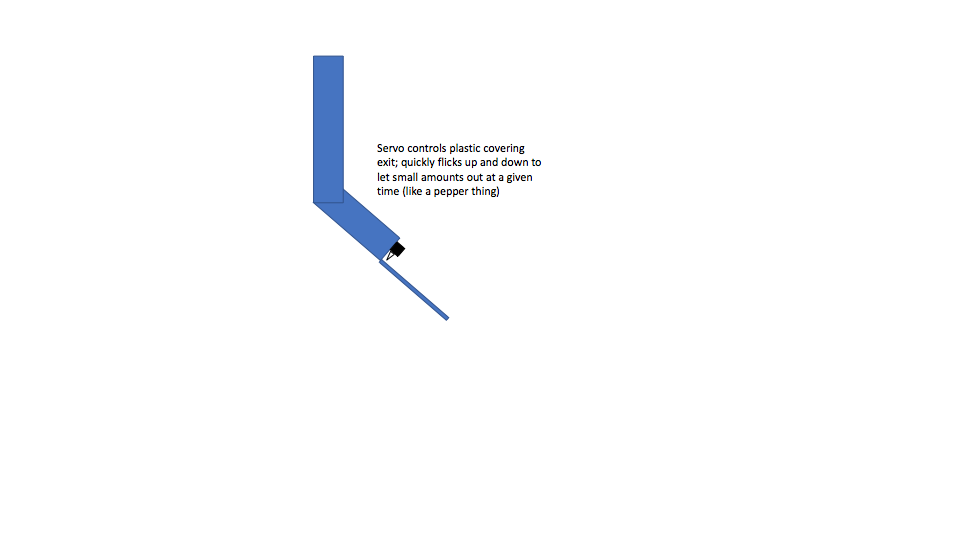


------

6/12/17:

Previous idea was difficult (coffee can with part of bottom face cut out, push coffee with dc motor). New idea:

Side-view:



So:

1. *Construct prototype to see if this works*
   1. *Take cardboard coffee cup for reservoir, use small piece of cardboard for flap (if not stiff enough, use small piece of plastic), get crappy coffee to use*
   2. *See how fast servo can flick between spots; we’ll probably want like 10 degrees or so*
   3. *Send a command to flick X number of times*
   4. *If successful, try to get idea of how many flicks = how much coffee*
2. *Construct real thing*
   1. *Material for reservoir... Could use coffee cans -> “funnel (pretty wide) -> cardboard cylinders? Plastic would be better, maybe old 2L soda bottles?*
   2. *Drill big hole in top of coffee maker and fix the setup. Need to figure out size of entry before doing. Remember that we want the coffee to come in at an angle.*
   3. *Make test program that flicks X number of times on command*
   4. *Run experiments to see how many flicks = how much coffee, make LUT to correspond to number of coffee cups*

**Filter Removal/Replacement Side:**

TODO: … well it’s all moonshot.

Honestly feels kinda impossible without complicated shit...current thought is:

* Have reusable filter secured in the filter slot of the coffee maker
* Have robotic arm with “hand” glued to front of filter
* Upon completed coffee, arm pulls filter out and flips+shakes it into trash can. Filter should stay bc secured, coffee grounds should be removed
* Robotic arm puts filter back

This is difficult and far beyond my knowledge of power and robotics. Would need higher powered motors and servos for sure due to the weight of the filter/holder and stuff. Maybe a future thing.

**Sleep Timer/Flag:**

Really just need to do this. Once the last part of the coffee making has been done, force it to wait for 1hr before being able to go again.

**Integration Side:**

Each individual side has to be integrated into the overall script. For dry software/hardware runs of complete script, do not have anything attached to its affectors (unplug the coffee maker, ensure water servo and coffee servo aren’t attached. I’m sure this’ll be a lot of troubleshooting and debugging. Make it happen cap’n.

One thing to worry about is the arduino and power/current usage...Make sure for the final script that each of the sections is complete before starting the next. Maybe some other time we can try and do everything in parallel.