**Description:**

It seems like it will be much easier to drive basically all the logic on our outputs from the Arduino and then report data to the Pi. In our final project presentation I think we can justify this by saying it would hypothetically allow for many arduinos to be driving their own seed starters, with one raspberry pi as the 'brain' handling everything else. I had issues with crosstalk over Serial between the pi and Arduino so it’s doubly beneficial to minimize that as much as possible.

My plan is basically that you work with the test files first. If we can get those working then (hopefully) everything else should flow pretty straightforwardly. If you need to adjust them let me know what changes you’ve made and I can make adjustments through the rest of the files if necessary.

**490ProjectInputTest:**

This is a .ino file which will test all the sensors and print to the Serial Monitor. If there's a sensor problem this is where we can debug it. Pins are declared at the beginning so that you can change them easily. In addition let me know if any math needs to be performed on our data to get it in the shape we want, we should definitely do that in this file.

**490ProjectOutputTest:**

This is a .ino file which will test all of our writes. If there's a problem with our outputs then this is where it can be debugged most easily. Pins are declared at the beginning so that you can change them easily. If a problem were to occur here I imagine it would be with the LCD. I have used much of our boilerplate code from previous assignments but wasn't able to test it myself.

**490ProjectFinal:**

This .ino file will hopefully just work if the previous two have. It integrates them plus it’s got the extra logic and everything contained inside.

**490ProjectNodeRed:**

Originally this handled a lot more, reading from a serial monitor, writing to a file, handling web and more. However when I went to run it on my raspberry pi I ran into a lot of problems installing packages. For some reason npm just refuses to work on my pi. This means I can't install Serial or Arduino or Dashboard nodes which makes the whole thing a bit more complicated. Can you try to run a few of these commands on your pi and tell me if they work for you?

npm i node-red-node-serialport

npm i node-red-node-arduino

npm i node-red-dashboard

If so we can cut out the python script I had to write below - I'll send you a couple node-red flows that substitute for the python script if you can install the serialport package.

As it is now it will still run a web server and although I don't have sensors I do have an arduino and pi. I have successfully been able to ping from the app and my web browser the node server which passes a command to my Arduino which my arduino then prints "Light On" and so forth. In the FINAL arduino code this is a digitalWrite. It also can send the entire csv we build in the python script below or just the most recent reading from the Arduino. The latter is how we update the data table in the app.

Because I wasn't sure if dashboard would work I haven't messed with that yet.

**490Project.py:**

This will need to be running before launching the node-red server but after you've pushed code to the Arduino. All it does is read from Serial. If it sees a comma in the string it reads then it appends it to a csv file. We use this csv file for the webserver and potentially for a dashboard. It's pretty buggy, I think because it's competing to read Serial. Either way, if we can do Serial through Node-Red it would've been much prettier.

**490ProjectAppInventor:**

This will send simple light on/off and fan on/off commands. After sending at least one command the table will pull data and repopulate every 5 seconds or so over wifi. I've tested it to the extent that it will actually update the table and that the on/off buttons do send a request.