Brian Wilcove

Santos Sanchez

1. Make a Buoy that will take pH readings and temperature readings, at both 1m and 2m, every hour. It will then send the data to a central repo and display the data on a webpage.
2. Goals
    a. Make a functioning website using flask
    b. Have a pi take pH readings and 1m and 2m readings of temperature and store the data if there's no wifi. If there is wifi, send it to a central rep
    c. Make a floating buoy that can house both the pi and its sensors.
3. Implementation: Make a floating buoy using this website guide http://www.instructables.com/id/BUOY/?ALLSTEPS
   After this we made a pi take sensor data and store it on a central repo that will take the data and create a graph of it.
4. /5. Buoy
    a. Make the buoy
        i. Took a lot of man hours but have a buoy that will not capsize under stress. More than ample room for the sensors we have now, could fit more on in the future.
    b. Make sensors take data and log the data within sqlite3
        i. We have all of the temperature sensors working.
        ii. We gave up on the pH sensor due to time constraints and trouble putting the data into a database.
    c. Send the data to the website
        i. We created a bash file that sends both the database and a csv of the database and sends them to a central repo located on our website.
    d. Make a website take the data and display a graph of the data
        i. We successfully pushed the data to the webserver and have a graph ready to read said data however we were unable to correctly parse the data and actively display it, so the current graph displays static variables collected from the night before the presentation.
    e. Attach the pi and assorted sensors to the buoy.
        i. We did not actually attach the pi/sensors however we displayed how we would have done so, and proved that our buoy had ample space to do so
    f. Run the system
        i. Running the system as a whole works fine up until the point where we need to actively display the collected data, so we used a placeholder graph to demonstrate how the data would be displayed
5. ^^listed above
6. Easy:
    a. While time consuming, the making of the buoy was pretty easy. Just needed the correct tools and the time to put it all together.

b. Code for the temp sensors is already basically done from what we did in class earlier. This does not count the pH sensor though
c. Working with flask turned out to be much simpler than previously thought. Very helpful framework

Hard

a. Getting the graph to read in our database was harder than we imagined it would be
b. Sending the files over to the internet didn't work on campus. We had to get an online server to make this work. We then had issues learning how to code in bash. Finally, in this process, we had issues getting crontab to work with bash because of directory issues.
c. Manual labor
d. Getting sqlite3 to work with flask
e. pH sensor
    a. Getting it wired correctly
    b. Getting the right equipment (missing a piece originally)
    c. Getting it to display correct pH's (had to recalibrate it)
    d. Lastly, getting the data out of the new terminal window that pops up (could not figure this out ☹)

7. People
    a. Brian
        i. Set pH sensors
        ii. Set up temp sensors
        iii. Worked with our local db file
        iv. Got materials
    b. Santos
        i. Set up local server
        ii. Set up online server
        iii. Set up chart
        iv. Web programming in general
    c. Both
        i. Made buoy (big)
        ii. Transferring files from pi that takes data to the web server
        iii. Setting up crontab
8. For the website:
    a. Used chartjs javascript resources from github
    b. Used bootstrap to learn basic html functions
    c. Watched many tutorial videos, in particular those from the youtube channel sentdex. I highly recommend them to anyone interested in starting out with python or Webprogramming'

Assorted others

a. Instructables
b. Stackoverflow
c. Sshpass
d. Tldp.org (bash programming)
e. Obviously your tutorials on blackboard
f. crontab