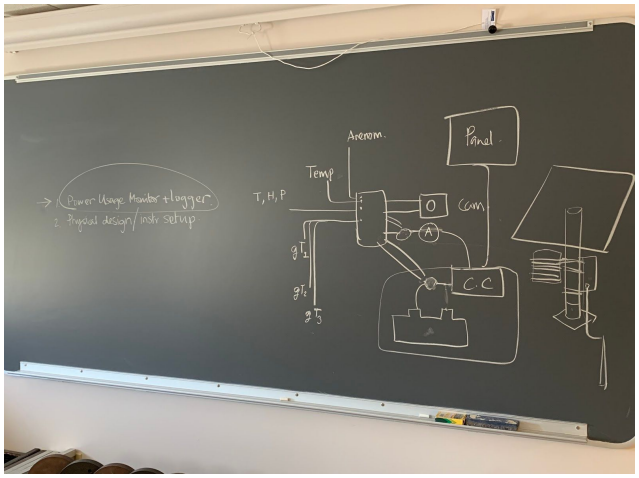
Sources used throughout the week:
https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/python-circuitpython
https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/wiring
https://www.youtube.com/watch?v=9LoMsYSCWTw
https://www.raspberrypi.org/forums/viewtopic.php?t=41849
https://raspberrypi.stackexchange.com/questions/13886/how-to-know-how-much-battery-power-is-remaining
analog to digital converter -
http://raspi.tv/2013/controlled-shutdown-duration-test-of-pi-model-a-with-2-cell-lipo
https://github.com/aboudou/picheckvoltage
https://github.com/kmcallister/rpi-battery-monitor
Very basic project - https://projects.raspberrypi.org/en/projects/build-your-own-weather-station/10
https://www.instructables.com/id/Raspberry-Pi-Solar-Weather-Station/
https://www.raspberrypi.org/blog/build-your-own-weather-station/
https://www.youtube.com/watch?v=kF_cVEYxj3E
https://www.youtube.com/watch?time_continue=212&v=j7LLVkPpQ78

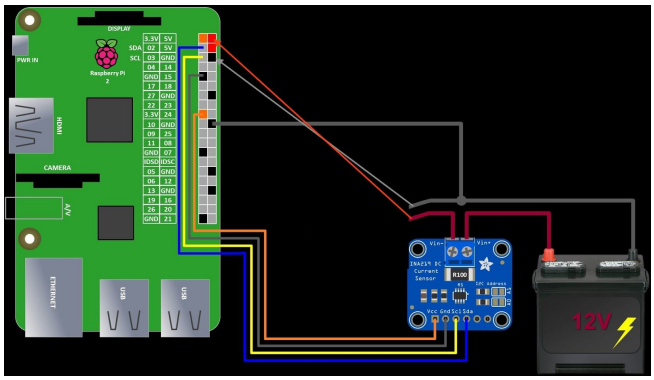
(very important link!) https://www.rototron.info/raspberry-pi-ina219-tutorial/

Monday, May 6

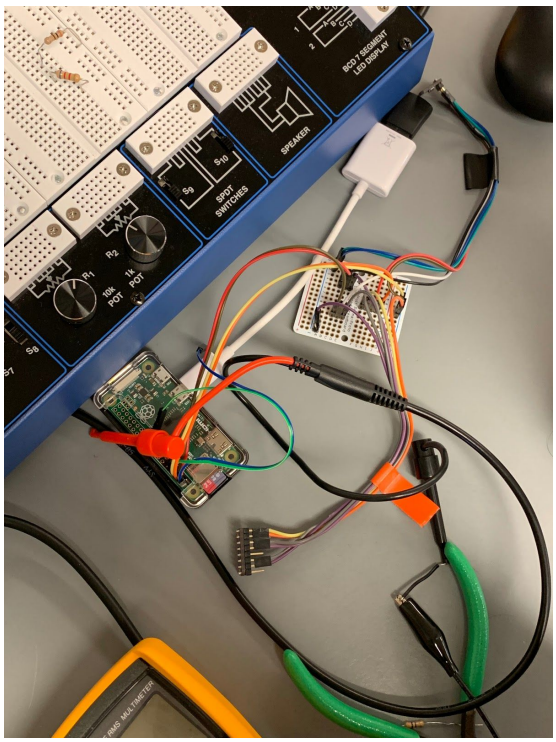
Over the weekend, Raj ordered the parts necessary to begin our testing of the measurement station. This includes the solar panel, the charge controller, the battery, additional wiring and more sensors for the raspberry pi. Today, we outlined the project on the chalkboard. I include a picture below. We realized that there are two primary parts of the project that remain.



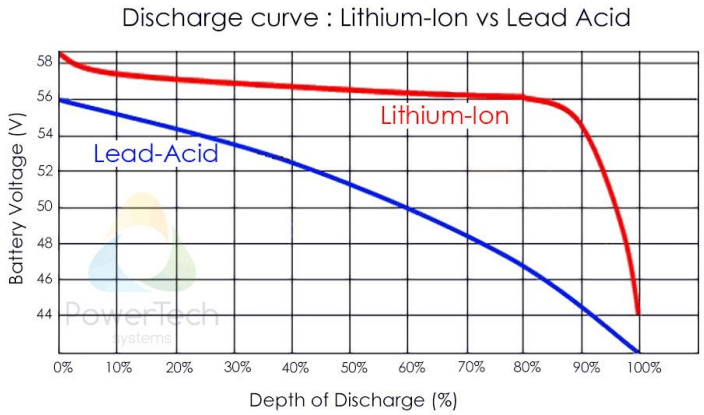
The first part is the physical setup of the station. The more important part is a power usage monitor. The power usage monitor we settled on is known as a INA219 (it actually measures current) and will be able to be logged on an SD card through some python libraries. With the INA219, we developed the below circuit with the raspberry pi.



Below, I simply include a picture of the raspberry pi that we are using and the two temperature sensors that we included to make our first power monitoring test.



For the remainder of class, our goal is to figure out how to monitor the amount of power stored in the battery. This is problem 2 on the chalkboard above. We want to find a way to measure charge level in the battery. To do this we are considering using an encoder to measure voltage or any other means to determine battery level. This concept is important because we want to know how much is left in the battery so we can determine data transmission frequency and we can determine if we need to turn off the raspberry pi. It would be extremely dangerous to have the charge controller stop operating and have the raspberry pi turn off without the appropriate shutdown commands. Below, we found a chart that shows the output voltage of a battery based on charge level. It is difficult to determine charge level directly, so perhaps we can even just measure output voltage and see how we can extrapolate charge level.



ATTINY13A-PU: this is the only part I think we need for measuring battery capacity. The resistors are variable and we are allowed to change them in the code. The larger the resistor we use, the less battery power we consume. However, he makes note that for a deep cycle marine battery, measuring battery level takes almost no energy.

Tuesday, May 7

Today, our goal is to research the python scripts that transmit data via the cellular network. We are going to research how others have made python scripts to transmit data via cellular networks. Furthermore, I am interested in researching other methods people have used to create these remote weather stations.

We turned on the raspberry pi using the USB and monitored it through a USB power consumption monitoring device. Raj then gave us instructions on how to log into the Python file such that we could rewrite the python scripts so the data is transmitted effectively in our autonomous weather station. Namely, we are trying to have to computer send data when there is ample power available and avoid sending data when there is not enough available power.

We were able to write preliminary python code (see the attached pictures). This code will help us reconcile the differences between transmitting data and storing data based on the length of the dataset. Eventually we will want to incorporate the amount of charge in the battery to run this code.

In [17]:
```
count = len(open("test1.txt").readlines(  ))
```

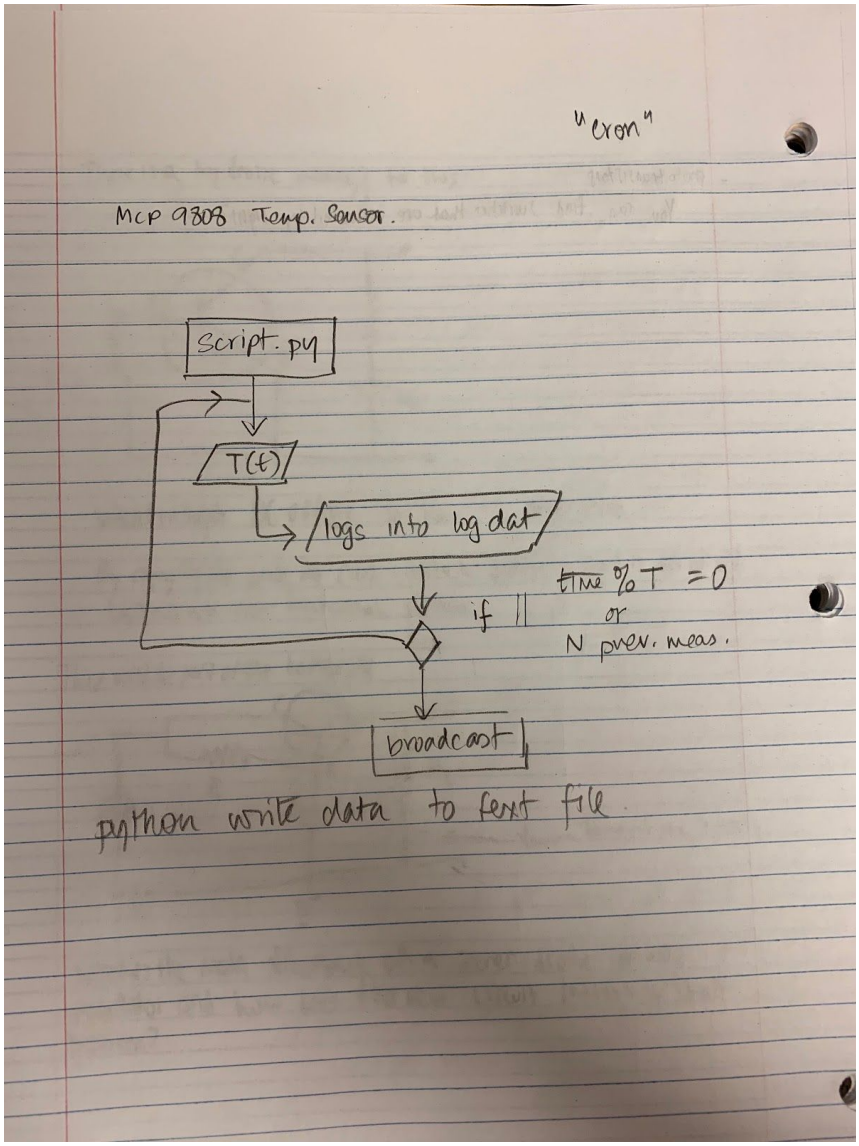
In [19]:
```
if count > 9:
    f=open("test1.txt", "r")
    if f.mode == 'r':
        content = f.read()
```

In [21]:
```
print(content)
```

```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
This is line 11
This is line 12
This is line 13
This is line 14
This is line 15
This is line 16
This is line 17
This is line 18
This is line 19
This is line 20
```

In [17]:
```
import numpy as np
f = open("test1.txt","w+")
for i in range(20):
     f.write("This is line %d\r\n" % (i+1))
f.close()
```

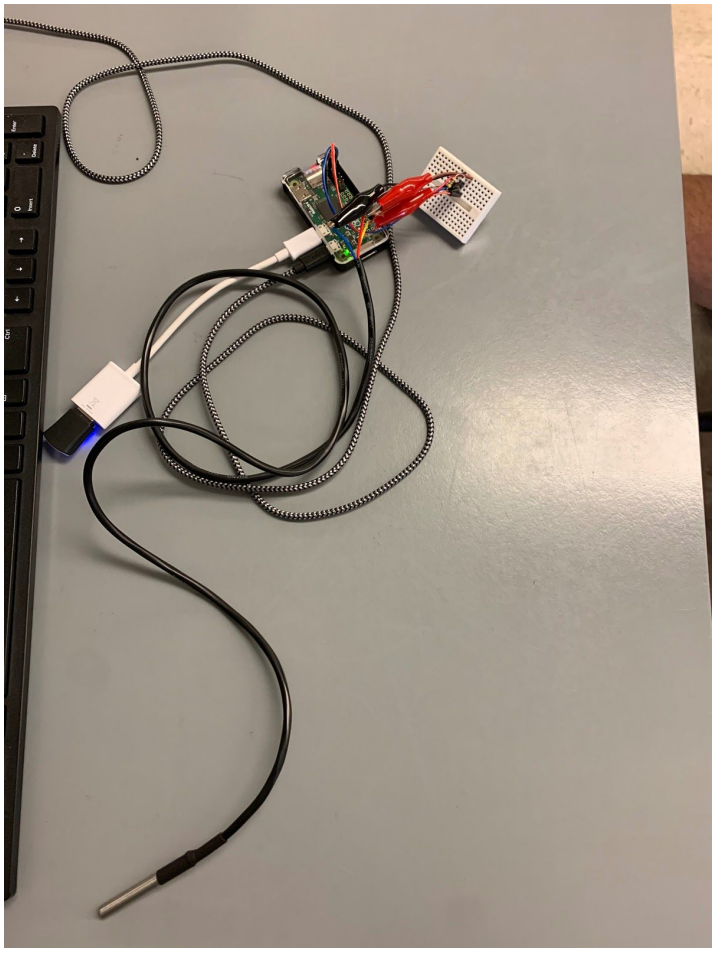
In [ ]:



Wednesday, May 8

We are setting up the new raspberry pi and the underground temperature sensor. THe three wires on the temperature sensor stand for VCC, ground and data.
http://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/
http://www.reuk.co.uk/wordpress/raspberry-pi/connect-multiple-temperature-sensors-with-raspberry-pi/

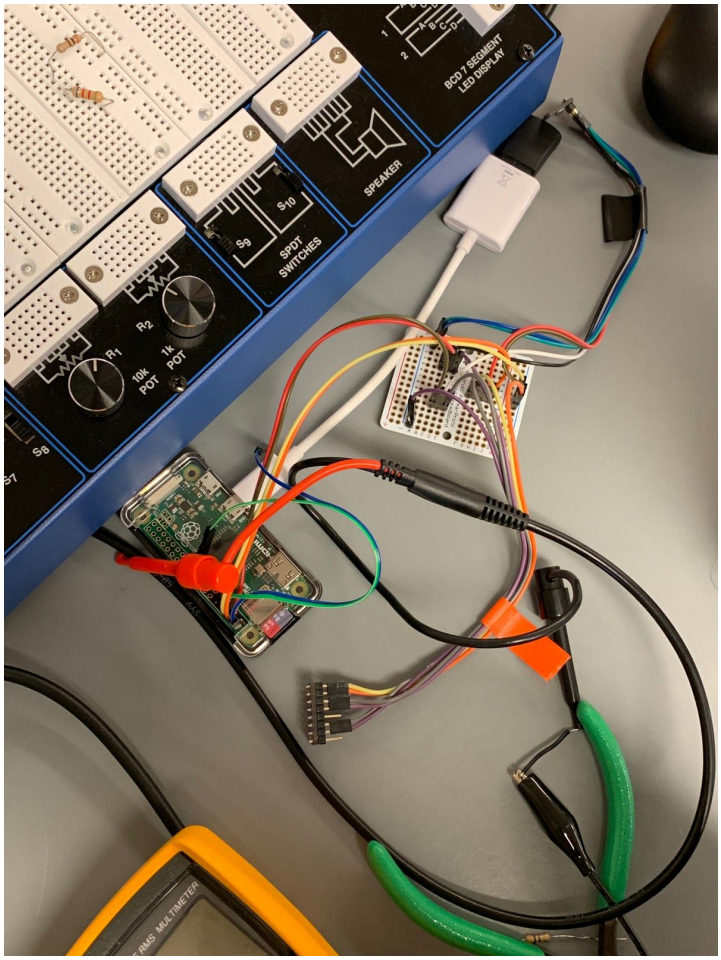
We have connected the temperature sensor to the new raspberry pi. Then, we connected the pi to the Bates network. Raj is now giving us the challenge to try to find the sensor address and then use python code to eventually record that data on the micro SD card.

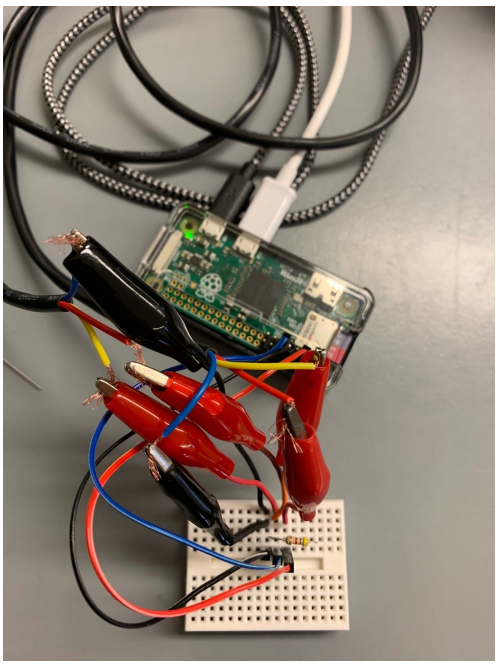


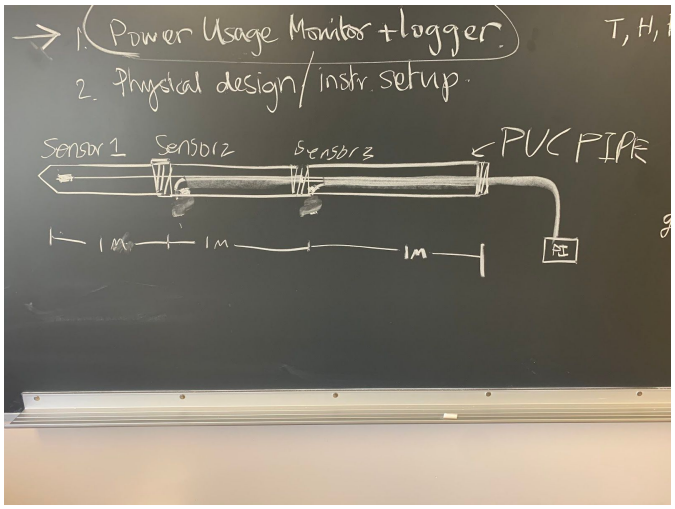
We got the temperature sensor connected and we have a python code that takes temperature and writes it to a text file every n seconds.

Now, we are going to connect two different temperature sensors in series so that we can take temperature measurements at various depths.

Here are a few other pictures from the day:



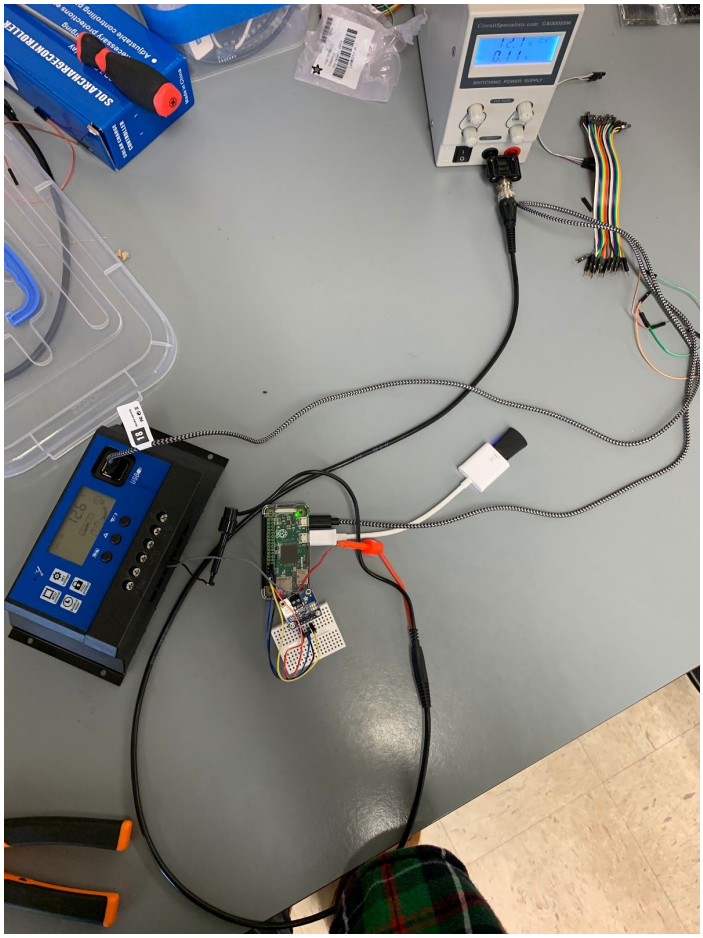




Thursday, May 9

The battery, charge controller and high side voltage monitor arrived today. The solar panel is a bit big so we are likely going to need another solar panel for practical purposes.

Today we are building the device that will measure the voltage output of the battery.



Going through the charge controller, we have powered the raspberry pi. We also have a sensor that measures high side voltage and current coming out of the “battery.” In our case the “battery” is a dc current generator. We have noticed that going through the charge controller significantly increases the amount of power that the dc current generator is outputting. We are soon going to test how much the 5 V usb output from the charge controller is outputting to make sure that we are not overloading the raspberry pi. We are a little concerned about how much energy we are consuming with the charge controller. We initially assumed 80% efficiency but it looks like the efficiency is more like 50%.

Ryan is uploading, via linux, the scripts necessary to turn on the sensor that measures high side voltage and current. These data will eventually help the pi make decisions about data transmission and operation when in the autonomous weather station.

For the future we also have to make sure to solder the current and high side voltage monitor such that things are stable.

Using python, we were able to take a reading of the voltage and current from the battery. We also soldered the pins into place for the sensor