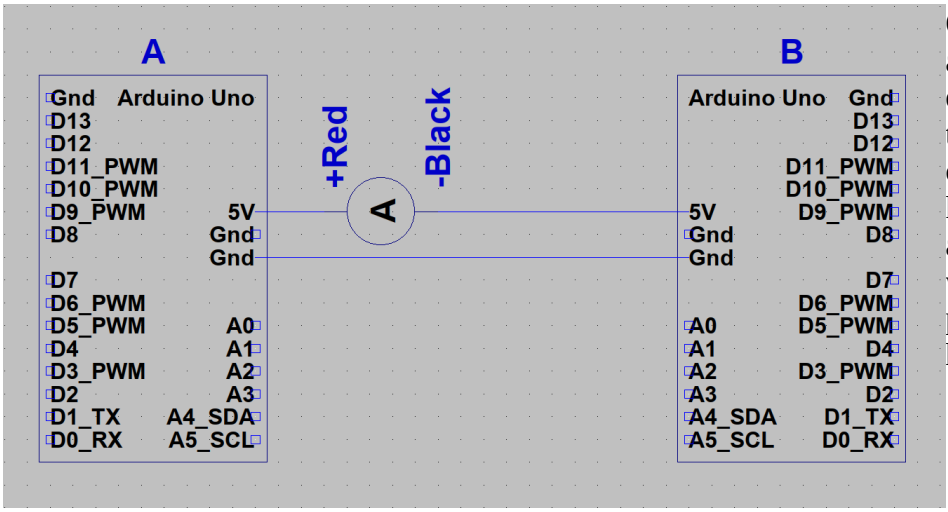
Let’s check to see how our system performs. We want to test our system power consumption and sensor responsiveness to make sure that all the sensors are still working fine after everything was integrated. We also want to make sure that our system will continue running for the entire 3 hour expected trip through the atmosphere. First, let’s address power consumption.

To test our system’s power consumption, we will need to pass the electrical current supplied to the payload circuit through a multimeter to be measured. This test would be ideally done by wiring a battery through the meter into the arduino Vin pin or through the barrel jack, this way the power measured by the meter will be using a power source similar to what will be launched. Our kit is minimal, so we’ll have go about this differently to estimate our systems power consumption since providing direct USB power offers no obvious way to pass the USB power through the meter before reaching the arduino and sensors. To measure our system power, participants will choose a neighbor sitting next to them to use one arduino to power the other through an ammeter. The second arduno circuit can then be checked to see if the two systems consume similar amounts of power.

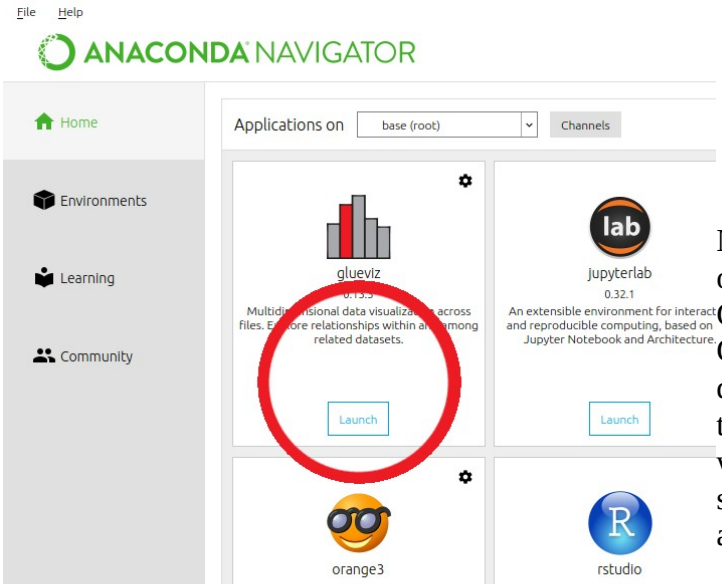
Construct the following circuit with arduino A only connected to USB power, and arduino B hosting the sensors and not connected to USB power. Arduino A’s 5V pin should be connected to the positive (red) terminal of the multimeter with it set to current meter and the negative terminal (black) connected to arduino B’s 5v pin to power it, with both boards ground pins connected together. This will ensure that any power that Arduino B uses will come from the arduino A 5V pin through the meter. To measure the system voltage, simply use the multimeter to check the 5v pin relative to the ground pin.



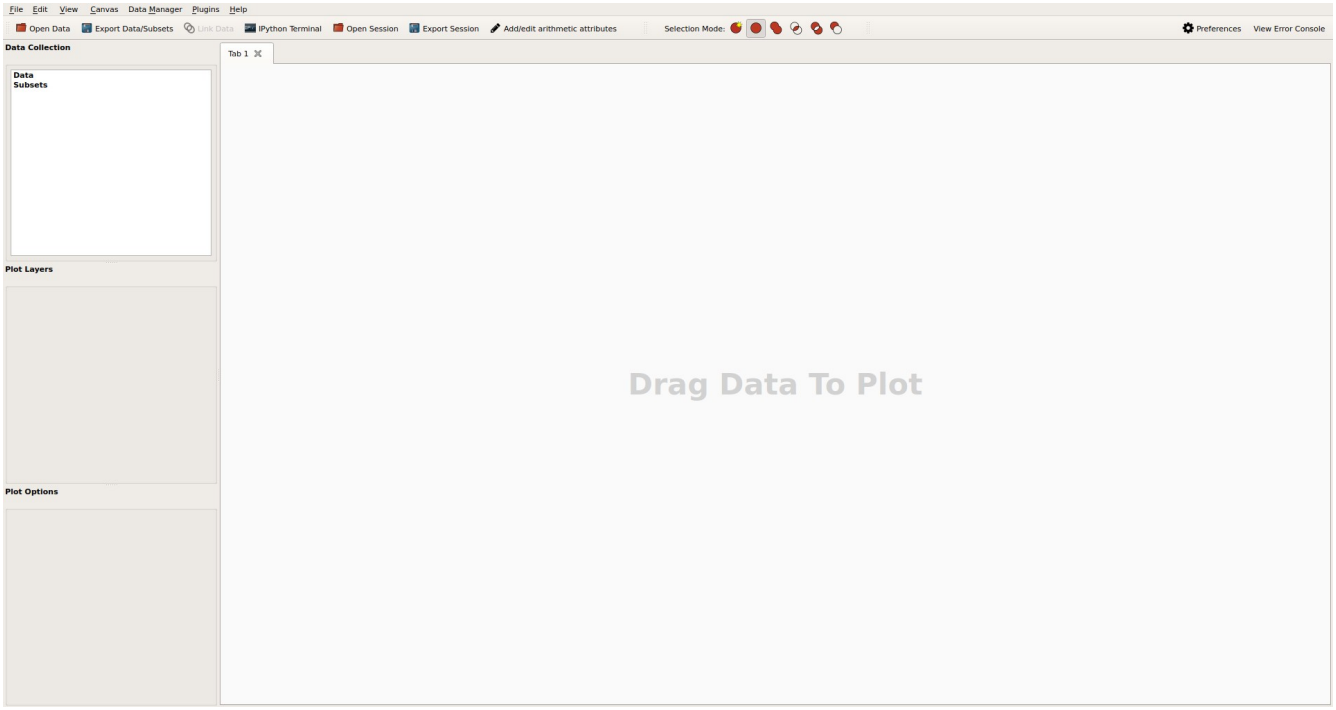
Often times USB power will actually be slightly below the desired 5V, so we can expect that if the regulated power coming from a battery can keep the 5V pin closer to actual 5V, then the system will consume slightly more power than we will measure here.

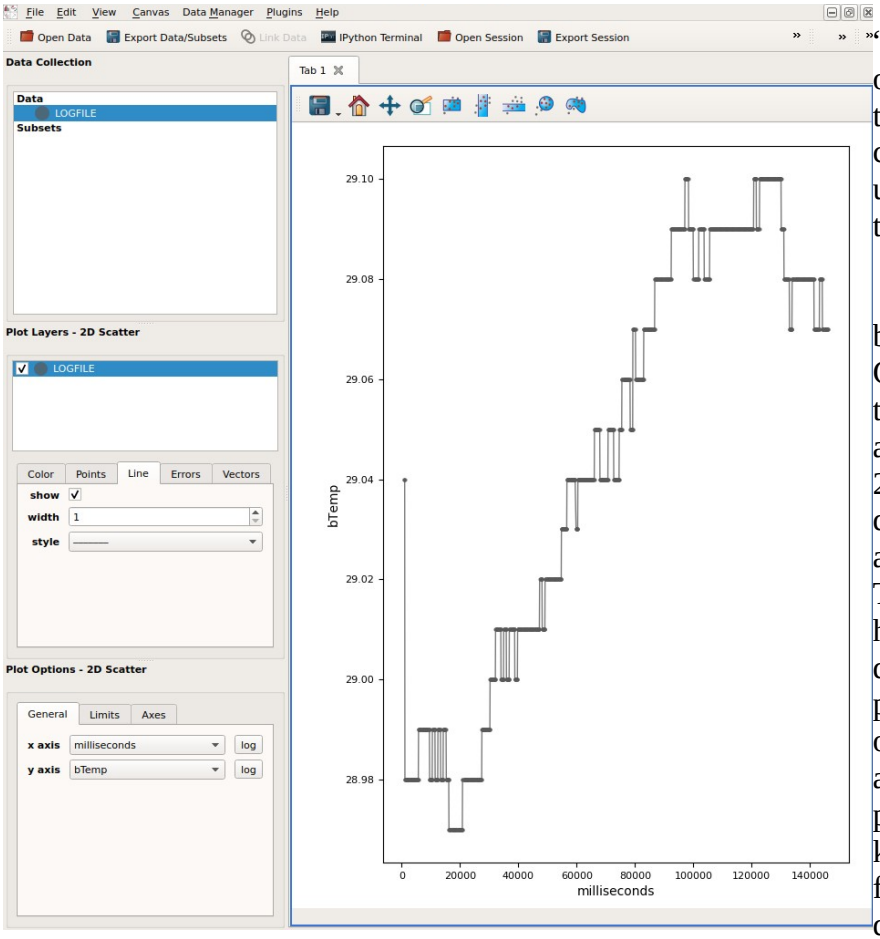
The system should be tested with an sd card inserted and with the system logging data and the activity led blinking so that the system activity will be similar to what we can expect from a launch. Write down the voltage and current you measure. After power consumption has been recorded and while we still have our system running, we want to check our sensors to make sure they are still working fine after we put the whole system together. We have a light sensor, air pressure and humidity sensor both with internal temperature sensors. With the system running and an sd card inserted, cover the photocell completely with your hand for a few seconds, and uncover it. Then slowly exhale onto the humidity sensor for a few seconds, and then do nothing for a few seconds to give it a change to reach normal humidity. Pressure can be difficult to test, so we can skip that step and instead simply place a warm portion of your hand onto the pressure sensor for a few seconds to warm its temp sensor, and the remove your hand for a few seconds to let it cool, and do the same steps to warm and cool the humidity sensor.

Now that we've recorded some test data, lets graph it to make sure all of the sensors responded to their stimuli. After the test was performed, disconnect power from the system and press the sd card into its receptacle to remove it, and insert it into your computer and copy the CSV text file to the computer. On the host computer, launch "Anaconda Navigator" and using the navigator, launch "Glue."



Here is a screenshot of the Anaconda Navigator and Glue once they have been opened. After Glue has been opened, drag the CSV data file from a file browser directly into Glue and the file will be imported and the dataset will show in the top left box. Then, drag the dataset from the top left box into the main window where Glue says "drag data to plot." A scatter plot will be created and the data on each axis can be set using the "plot options" area.





Set the x axis to "“milliseconds” to graph all of the other values against time, and a line to connect the dots can be set by checking the “show” checkbox under the “line” tab on the left of the application.

Here is a view of test barometer temperature in Celsius. Check your sensor data to see that the temperature sensors report ambient temperatures of around 25*C and have increasing and decreasing regions to correspond to a warm hand touching the sensor. The light sensor should also reflect high and low regions for bright and dark readings as well. Since pressure wasn’t stimulated, we can only check to see that the value is average ground level ambient pressure, which is around 98 kiloPascals. If all sensor data looks fine, we can proceed to take a deeper look at system capabilities.

An example system was tested to be an average 45milliAmps at around 4.8V.