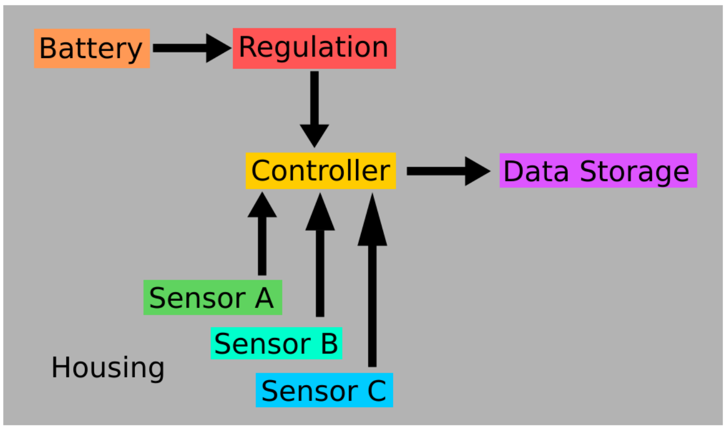
Now that we have some background information on ballooning and electronics, lets begin designing an example payload.

A proper experiment should ask some question, and determine what data must be recorded to answer that question. If no prior experiments have been done, a great initial payload experiment is to measure some basic values to get a general idea of what the upper atmosphere is like. Some straightforward values that can tell us about the atmosphere are air pressure, temperature, humidity, and a qualitative light measurement can be used to measure some context like rough orientation relative to the sun. A light sensor facing upward on a payload with the sun high in the sky would read highest with the payload oriented upward and would read some lower level if facing horizontal, and from this information we can determine qualitatively what could be happening to the payloads motion.

The following is a generalized block diagram for a payload and its subsystems.



We've decided we want to see how air pressure temperature and humidity vary based on altitude, with an extra qualitative light measurement, and we need a method for storing said data. For now, we'll assume using an arduino uno and the power source we specified earlier, but we will examine other controller options later for more strict requirements.

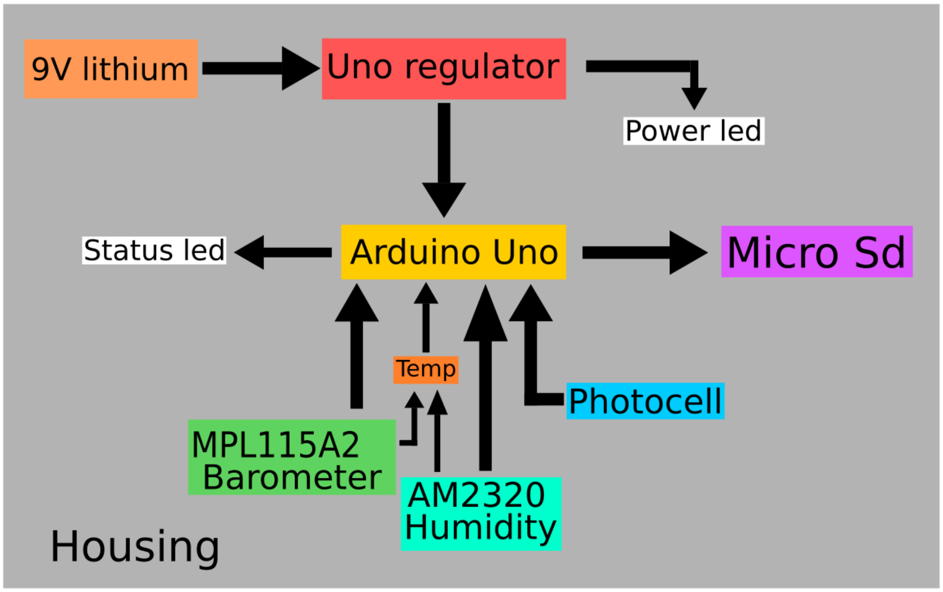
Our payload is for demonstration purposes only, so our sensors will be chosen for their low price point, and as such are not actually rated for extreme temperatures and low pressures. But they will still be sensitive enough to be able to produce general trends. In a properly executed payload, the sensors should be chosen sensitive enough to be able to measure the target values within the project budget. For example, if the desired phenomenon to be measures is some small temperature variation as small as 1 degree C, the temperature sensor used should be able to detect variations a good deal smaller than that, since that variation would be effectively invisible to a sensor that can only detect 5 degree C steps. In the same way, data should be logged at an appropriate rate such that the system can see the target variation. For example, if the event to be measured only last 30 seconds, but if the system only logs data once per 45 seconds, it will be easy to miss the desired data. However a data point logged once per 5 seconds could be enough to see the event.

We've previously specified the use of a lithium 9V battery and the arduino's internal regulator for power regulation. We can use a snap to barrel jack connector like the one here as a simple solution.

If we want to do some general measurements just to get a basic idea of what the upper atmosphere is like, we need to correlate the sensor data logged over time. We will want to be able to at least graph the individual sensor data over time, and we'd want to be able to be able to correlate each sensor against each other to see any dependence of one value on another. For example, graphing air temperature depending on altitude would be a useful correlation, which can be indicated through a pressure sensor data. Some sensor data may not be valid for all regions however, for example data read when the sensor is outside its rated temperature range, or if some gas sensor requires a minimum air density to operate accurately.

We want to measure air temperature, barometric pressure, and humidity, and a qualitative light sensor to determine if the payload is pointed up, and then that sensor data with a timestamp must be stored. To satisfy these requirements, we will use an MPL115A2 barometer, AM2320 humidity sensor, and both the barometer and humidity sensor have internal temperature sensor with different sensitivities, a photocell for light sensor, and a micro sd card breakout will be used to log this data to a text file on a micro sd card. We also want to be able to visually see the system is operating properly with no host computer connected to it so any system issues can be detected prior to launch, so we will use 2 indicator leds, one to show power is connected, and another to show system activity.

With this new system information, our general block diagram becomes:



We will organize the data in a text file on the sd card with the values separated by commas and including a timestamp generated by the arduino, with a leading line to label what each comma position represents. So our leading line, called the header will be the following, and our final system will be made to follow this order.

“baro,humid,baroTemp,humidTemp,light,timestamp”

From this, in our data analysis stage can then calculate approximate altitude based on pressure readings, or average the temperatures from the two sensors, but we will simply compare how sensitive they are. We will use excel for data manipulation, and a program called glue to graph the data.

Now that we have a plan for our electrical system composition, we can go on and test every sensor to make sure each one works, and then we can finally assemble the electrical system.