**Team Name**: Honey Badgers

**Date**: 11/5/2016

**Team member name and ID**:

John Walter

Thomas Dye

**Target accomplishment by the report date**:

- Xbee radio transmission between controller and command station. The controller can wirelessly send analog test data from sensor probes to a PC-based command station. The PC-based command station can send instructions to the controller through a rudimentary functional API.
- Final hardware requirements are identified and scheduled for procurement.

**Actual accomplishment by the report date**:

- We are currently able to transmit ASCII messages over XBee 900MHz radios in API mode. We currently have functional code to send messages from an Arduino to a Linux-based server, and vice versa. We have not drafted an API or command syntax yet, however.
- We have written individual sketches to activate and test all of our sensors applicable to this project, to be runnable on Arduino. The sketches currently exist as separate sketches to verify functionality and illustrate API calls.
- We have obtained the last of our major components for this project, but smaller subcomponents (such as support materials for powering electronics and materials for physical structures like the sensor stack) need to be cataloged and purchased.

**Analysis of the progress**:

Working with the XBee radios has proven to be very difficult and has consumed huge amounts of time. We do not have the right kind of hardware to debug the radios and examine the frames directly, and the API library documentation is almost non-existent. Most work is accomplished by trial and error where each complex task (i.e., transfer the contents of a string) is broken down into multiple smaller tasks (i.e., establish a connection. Then send a byte. Then send multiple bytes. Then reassemble them as a character array), and this is very slow. Often, something fails and it's not clear if it was the sender or receiver code, or the radio configuration, or the API doesn't work.

Working on two entirely different platforms using two different libraries also greatly increased the complexity of this. The Arduino library wants the radio to be configured in a way that does not work with the Python library we're using. The Python library has even less documentation and support than the Arduino library. More trial and error programming, and no way to intercept the packets to decode them manually because we don't have the hardware to do so.

This week we will evaluate what, if anything, needs to be cut from the project. We also need to figure out how this will be presented, as this project does not have a good stage presence. By that, I

mean the only physical change that can be seen is that a valve will open (it can be heard, but not seen) once an algorithm decides it needs to and that action is scheduled.

There are still several large hurdles to overcome, given that both of us have light or no experience with Python, RESTful APIs, and web services in general.  The amount we cut is dependent on the amount of time we can dedicate to learning foundational knowledge to accomplish what we want to do.

**System picture**:

There are no significant changes in our hardware setup this week since we have just been working with library code.  Some new hardware is pictured below.