**Initial Concepts**

*Function 1: Control the shed environment*

1. Have sensors for each of the conditions that need to be controlled (i.e. temperature, humidity, CO2).
2. Write a Python script for the Raspberry Pi to switch relays on heater, humidifier, or CO2 emitter as directed by the sensor readings.
3. Have the software trigger alerts via text or email when the controller fails or conditions drop below a certain threshold.
4. Build housing for the RPi or PC to prevent any damage. Maybe mount housing to the reactor or wall to conserve space.
5. Or we could scrap the controller idea and just use timers to control the heater, humidifier, etc.
6. Switches could also be controlled manually, either by physical switches in the shed or remotely via ssh into the pi.
7. Raspberry Pi could have a touch screen interface with GUI to control switches.
8. System could also include a light and motion sensor to detect when somebody enters the shed and then turn on the light.
9. Use something like the BeagleBone instead of the RPi if we determine we need more capability.
10. The controller should be configured to avoid short cycling. It should have a few degrees of hysteresis.

Lots of examples of people doing similar projects with RPi: <https://www.raspberrypi.org/forums/viewtopic.php?t=197477>

Example pseudocode:

|  |
| --- |
| while(true)  sensorReadHeat()  sensorReadHumid()  sensorReadCO2()  if sensorHeat is low:  turnOnHeater()  if sensorHumidifier is low:  turnOnHumidifier()  if sensorCO2 is low:  turnOnCO2() |

*Function 2: Record & present shed environment data*

1. Run program to automate recording via Raspberry Pi.
2. Store the data locally on the Raspberry Pi or another PC in the shed.
3. Push the data to RIT servers and make data accessible remotely.
4. Have 2 monitors in the shed. One to see shed conditions in real time and the other to see historical trends.
5. Store the data in JSON files or SQL database.
6. Extrapolate data into visual graphs that show real-time and historical data trends on the monitors in the shed.
7. Use software like PowerBI or Thingsboard to present visualizations so they would be available remotely.
8. How often to push data? Report more frequently for prototype purposes (maybe every 2 min) but not as frequently when the control system is more finely tuned.
9. How far back do we need data? Overwrite old data after a certain file size limit is reached. This limit will depend on how much storage space is available to us.
10. In addition to shed environment variables, we would also record data on the solar system to better understand energy usage/requirements.

Example Dashboard:

