Website Design Document:

The website will do the following:

Receive data from the main Raspberry Pi (in JSON format, or a ZIP format if it’s too big or something)

Process that data with these steps:

1. Store this data and its timestamp into a MongoDB server.
2. Update the instantaneous JSON data file with this new piece of data.
3. Update any graphs that use this data.

Example:

Receiving a POST request containing MPPT Data on 11/27/2019 @10:00am:

1. Store the data to the MPPT database. Example columns: Voltage, Power, Current
2. Update the previous instantaneous JSON data file with this one. Now when you open instantaneous.html, you will see the data from 11/27/2019 @10:00am, instead of the data from 11/27/2019 @9:00am.
3. Update the graphs that display voltage over the past 24 hours (example graph). Shift all data points to the left, eliminate the leftmost datapoint, and add this data point to the right.

Will be a server, too. If someone using the eduroam Wi-Fi network tries to connect ( I haven’t set up port forwarding yet, so you’ve gotta connect to the school’s wifi network to reach the server), it will ask what webpage it’s trying to get. The browser will automatically tell the server what it’s asking for.

Index.html:

Returns a webpage asking what kind of data you’re looking for.

* Tristar Morningstar MPPT Solar Power Tracking Data
* Emus BMS Monitoring Data
* Experiment 1 Data
* Experiment 2 Data
* Experiment 3 Data
* Experiment 4 Data
* Experiment 5 Data

Tristar.html:

Returns a webpage with the instantaneous data on it, and a graph of the past 24 hours. If a new instantaneous value is generated, it will automatically refresh itself to include that data.

It will also have a form that can update the instantaneous data values on the page to whatever they were at whatever time and date you want. If the exact time and date does not exist in the server, it will return the values with the time and date closest value to your requested value.

Morningstar.html:

Returns a webpage with the instantaneous data on it, and a graph of the past 24 hours. If a new instantaneous value is generated, it will automatically refresh itself to include that data.

It will also have a form that can update the instantaneous data values on the page to whatever they were at whatever time and date you want. If the exact time and date does not exist in the server, it will return the values with the time and date closest value to your requested value.

Experiment\_1.html: (2,3,4,5)

Returns a webpage with the instantaneous data and its names on it for each respective experiment, and a graph of the past 24 hours. If a new instantaneous value is generated, it will automatically refresh itself to include that data.

It will also have a form that can update the instantaneous data values on the page to whatever they were at whatever time and date you want. If the exact time and date does not exist in the server, it will return the values with the time and date closest value to your requested value.

Writing a website:

I will have to ask the UCSC IT guy to set up port forwarding to this server. I don’t think I have proper permissions to do that. And to see if my flask server is acceptable to assign to a UCSC name.