For my project of the external hardware monitor I have acquired all the hardware that I need. This includes the Pimoroni HyperPixel - 4.0" Hi-Res Display for Raspberry Pi non touch version and a Grove Chainable RGB LED. I was able to install the screen onto the pi and install the software. Since it is not HDMI it is not plug and play. This is frustrating because the software disables HDMI and locks the screen resolution to 480p even when using VNC. The only way around this is to either uninstall the software or load a different microSD card into the pi, I chose the latter. So, all my software development has been done and tested on the same pi with the screen still physically installed, but without the software for it.

I had a very slow start with development. Being new to python I had a hard time jumping right into something more than basic. My first issue was that I was convinced that I needed to scrape the website itself to get ahold of the data I needed, something I’ve never done before either. After much trial and error, I finally figured out that the data wasn’t even on the website, and that there were java scripts that were displaying everything. I felt like this was the end of the road, if I was not able to see any of the data by viewing the page’s source code then this project didn’t seem possible. More reading showed that I needed to turn my attention to the json file that I found by going to the network tab of the browser source code viewer. This is the file that had everything I needed, only in a strange format I had never experienced before. After learning how to scrap data from a website I learned how to read data from a json file. The formatting was different that most examples online and it turned out I need to read it using recursion because of the nesting structure. This gave me a Unicode list/string of everything that I saved to a text file.

Once I could get my hands on the numbers that I needed I felt a lot better about my progress. At this point it had been around 15 hours and I finally had something useful. Quickly I figured out how to pick out specific attributes that I wanted, like CPU package temp instead of separate temperatures for each core. I put everything in an infinite loop with print statements and a 1 second sleep. This means I have a terminal window that displays data and updates once per second by requesting the website each time. Next, I figured out how to scrub the % and °C from the Unicode using regular expressions. This allowed me to convert each entry into a float type to be used for graphing or other purposes. I have only just started using matplotlib to create a simple graph, but it seems promising.

I have not tested functionality with the LED yet but GPIO programming seems to be very easy in python. I did plug it into the grove port on the Pimoroni and its PCB got extremely hot, so I am considering using a plain LED instead. I was planning on having the RGB LED color shift depending on temperatures but an off, solid, and blinking led could serve the same purpose if needed. From here I will continue to add graphing and then an acceptable GUI to contain everything. Once that is done, I will add the LED to the mix and should be close to finished. The final thing that I want to add is a config file where a user would be able to decide what information is displayed, what order, and how often it is updated. Due to the nature of the json layout most computers will not have the same data order. Currently it is read by what line number I know needs to be used, I may be able to change this to recognize specific components. This would make the user setup easier.