Working on the development of this embedded system throughout this semester has been an up & down rollercoaster of a struggle. There have been ample opportunities to explore the integration of multiple hardware components, like a temperature sensor, a Led display, and some questionable user input through a button. While the project's initial design phase proceeded smoothly, unforeseen complications arose during implementation, particularly with the LED display module. These challenges not only stalled progress but also provided valuable insights into the complexities of hardware interfacing and the importance of robust system design.

The thermostat was intended to be the big bang at the end of the semester. Instead, it led to more frustrations, as I was unable to overcome even after revisiting the previous modules. The sensor when working correctly, should be capable of reading ambient temperature, displaying real-time data, and accepting users to adjust settings through physical buttons. I believe the script that I had designed for the sensor tended to be mostly correct and executive in visual studio without flaw but could not get it to display on the led.  The GPIO-controlled buttons and status LEDs operated as expected, responding to user input and indicating system states appropriately. The state machine logic, which dictated whether the system should heat, cool, or remain idle, functioned reliably in preliminary tests. The project just seemed to get hung up on my LED not displaying which caused me to take some drastic measures as I needed it to proceed.

The LED, as critical as it is, has problematic from the outset. My 1st attempts to install it weeks ago lead to a lot of second guessing & restarting the assembly process. While initially it did not power on, I was able to coax some feedback from it, thinking it was successful. Despite these attempts, looking for outside sources of help, a YouTube video & some breadboard guides, I was able to power the backlight, set up my contrast control, and change the output. So, I knew I was on the right path. Later in the semester, towards the home stretch, my troubleshooting induced an error I could not overcome, which was the shorting of the LED circuit, permanently damaging it. This hardware failure brought the project to an abrupt halt, as the screen was essential for verifying system functionality. Without it, the thermostat could not provide the feedback needed, rendering further testing impractical.

One thing I can take away from this is understanding the hiccups of the development process in products. Hardware reliability is a major point while also the weakest link. Software can be debugged, patched, and essentially fixed after product release. Physical components are susceptible to permanent damage from minor mistakes. There may have be alternative ways to continue the process via a secondary method. Though that may also have added more headaches. With other modules, it is essential to have a debugging process, like testing the light up bulbs from earlier modules. Verifying electrical connections would have saved much time.

Beyond these troubleshooting steps, perhaps a different selection of hardware might have been better to use. A web browser-style server could have provided remote access to the system data, removing the need for the on-board LED to display information. A backup alternative, if a bit uglier.

In conclusion, though I couldn’t complete my project, I did enjoy working towards the finish line. Plenty of ups & downs, enjoyment found in small success I found week by week. This let me know how important it is to have a back up plan & good debugging.  Moving forward, I intend to apply these lessons by incorporating redundant output methods, implementing stricter hardware safeguards, and adopting a more methodical approach to troubleshooting. While the project did not reach its full potential, I enjoyed the journey. This just lets me know in the future, work better with software than hardware, be resilient & diligent. Maybe a bit more patient when it comes to troubleshooting.

joshi, V. (2023, October 4). *What are some of the best chips for embedded systems?*. Medium. https://medium.com/@vedjoshi03/what-are-some-of-the-best-chips-for-embedded-systems-77f384c3eac1