**Name:** Scott Cain  
**Course:** CS350 Final Project  
**Date:** 6/22/25

**What I Built**

This project is a working smart thermostat prototype running on a Raspberry Pi 4B. It reads the room temperature using an AHT20 sensor, shows the info on a 16x2 LCD screen, and uses LEDs to show whether it’s heating or cooling. You can press one button to switch between off, heat, and cool, and two more buttons to raise or lower the set point. The system also sends status updates over UART, which acts like it’s pushing data to a remote server.

Everything runs through a Python state machine. Depending on the current state and temp, it turns the appropriate LED on or pulses it if heating or cooling is active.

**State Machine Design**

There are three states: off, heat, and cool. The green button cycles through them. Each time you enter or exit a state, the system updates the LEDs and display. For example, entering heat mode triggers the red LED logic, and exiting cool turns the blue LED off.

**Moving to the Cloud: Choosing the Right Hardware**

The prototype was built on a Raspberry Pi, but for production, the thermostat would need a Wi-Fi-connected embedded device that runs this logic more efficiently.

So I looked at three hardware options: Raspberry Pi, Microchip (like a PIC32), and Freescale/NXP (like a Kinetis). Here's how they stack up for building a smart thermostat that talks to the cloud.

**Raspberry Pi** is what I used for the prototype. It worked great because it has built-in Wi-Fi, plenty of GPIO pins, and enough memory to handle everything I needed—including future updates. Super easy to program, and I didn’t have to fight it to get the sensor or display working.

**Microchip** boards are more low-power and definitely more “embedded.” But most don’t come with Wi-Fi built-in, which means I’d need to add external hardware just to get it online. Also less memory, which could be a problem if the software grows or starts doing more analytics.

**Freescale (now NXP)** feels like a solid middle ground. Some of their boards have the right features, but not all of them come with Wi-Fi built-in either. More flexible than Microchip in terms of memory and resources, though.

**Bottom line**: I’d stick with Raspberry Pi for development and early cloud testing, then move to an NXP board with built-in Wi-Fi for production to keep power and cost down.