I've finished building the first version of our smart thermostat prototype, and it's working just as we planned. The goal was to prove we could handle the basic functions, and I can confirm that we can. I wrote the code to be as clean and straightforward as possible, with plenty of comments, so anyone on the team can jump in and easily understand how it works. The program's main logic is built around three simple modes: off, heating, and cooling. A main button lets you cycle through these modes. When the thermostat is in heating mode, a red light turns on; it gently pulses to show it's actively warming the room and stays solid when the room is at the right temperature. The cooling mode works the same way but with a blue light. Two other buttons are used to raise or lower the target temperature one degree at a time. All this information, like the current time, temperature, and what mode it's in, is shown on a small LCD screen. To get ready for the next step, the device is already sending out a status update every 30 seconds over a serial connection, just like it would to our servers.

Now that the basics are working, we need to decide on the computer chip we'll use for the final product that we sell to customers. The decision comes down to three main things: the chip has to be able to run all the parts of our circuit, it must be able to connect to the internet to send data to our servers, and it needs enough memory to hold our program and run smoothly. I looked at three different types of chips to see which would be best. The first is the Raspberry Pi, which is the small computer we used for this prototype. It's great for testing because it has Wi-Fi built-in and can handle all the parts of our circuit easily, but it's way too powerful and expensive for what we need in the final product.

The second option is a more traditional, basic computer chip from a company like Microchip. These chips are great at controlling the circuit's parts, but getting them online is more complicated because we'd have to add a separate Wi-Fi module, which makes the final product harder to design and build. It has just enough memory for our current code, but we might run into limits later on. The third option, from a company like Freescale or NXP, seems to be the perfect middle ground. These modern chips are powerful enough to handle everything we need, and many of them come with Wi-Fi already built-in, which is a huge advantage. They have plenty of memory for our current needs and for any new features we want to add down the road.

Based on this analysis, my recommendation is that we build the final thermostat using a chip from the Freescale/NXP family. It hits the perfect balance of cost, power, and simplicity for our project. It's not overly expensive like the Raspberry Pi, and it's not as complex as the Microchip option since we don't have to worry about adding a separate part for internet connectivity. This choice will give us a solid and flexible foundation to build a really great product for our customers.