Jacob Englund

Professor Morales

CS 350

19 April 2025

Thermostat Report

The goal of this project was to create a simple concept of a thermostat. The basic model uses LEDs, buttons, and an LCD display to demonstrate heating and cooling of a given room. When the system is turned on, it immediately is put into heating mode. Here the red LED is turned on and will either blink or be solid depending on the set and current temperatures are. For example, if the room is colder than the set temperature is, the LED will blink indicating it is actively heating the room. If the current temperature is above the set temperature, it will remain solid indicating no heating occurring. Another button click turns the system into cooling mode. Here it is the opposite of the heating mode. If the current temperature is above the set temperature, the LED will blink indicating cooling. The LED is solid if it is the other way around due to cooling not being necessary. One more button click turns the system off. There is also an LCD display that says exactly what the system is doing. The data is also being output via a USB output. This data can be stored on a server to show heating and cooling trends in a given time period.

While the Raspberry Pi works great for this application, it is a little overkill. The Raspberry Pi can handle much more intense tasks making it useful for something else. Using a smaller processing unit could also reduce the size of the thermostat and require less space. A lot of other chips would be able to handle the peripherals as well. Enabling Wi-Fi would also be an amazingly easy step for a different unit. A lot of different systems like the Arduino for example can be fitted with a Wi-Fi adapter to allow it to connect to your home network. Since the system is basic right now, it is hard to say how much ram and storage the device would need. I do believe that this number would be ridiculously small, and a custom chip could be made to support this requirement.

Currently, the Raspberry Pi that the thermostat is running on is directly connected to the Wi-Fi. This means that all data being collected by the system could be transferred to a cloud service that can keep track of it. This could include a server to store data or a mobile app that can be used to remotely control the system. Different architectures can also be used like this. Allowing a device access to the local network gives it tons of access to neat features like this.

The code used to run this current model is Python. There are a lot of different architectures that support this language and could even directly support this code. The Raspberry Pi does use a GPIO numbering scheme that will be different than another architecture. This is easily fixed though because the code is very modular. I have no concern that any architecture could be used for this system.

In all, the Raspberry Pi thermostat is an excellent prototype for the smart thermostat product. I think with a little tweaking; it could be easily used by multiple different architectures. Even though the Raspberry Pi is not the most efficient one for this system, it does work very well and is already running a working version of the system. Factoring in things like cost and waste, it may not be the most perfect choice, but it would be a valid choice for the final product!