Brian Engel

Thermostat Project

For the thermostat to work as intended any microcontroller we use has a list of peripherals that it needs to support. First it needs to support I2C and have a temperature sensor or be able to connect one. It also needs GPIO for a LED light and two GPIO interrupt buttons to increase and decrease the temperature. In addition to that it needs to have an integrated timer. Last it needs UART for an output of the current temperature, set point for the temperature, indicator of heater being on or off, and seconds since reset. The microcontroller that we used in the class, the TI SimpleLink Wi-Fi CC3220S development kit (Texas), was able to handle all of these requirements. The Microchip WFI32-IoT Development Board (Microchip) also seems to have these capabilities. I couldn’t find a board from NXP that had all the requirements, specifically the temperature sensor. This isn’t necessarily a deal breaker though because the FRDM-K64F development board (NXP) has expansions so you can add one, although that would add cost.

To connect our board to the cloud, we need the microcontroller to have Wi-Fi capabilities. We will use Wi-Fi to connect to a network and use communication protocols to connect to a cloud service over the internet. This will allow us to control the thermostat from anywhere. The TI board has integrated Wi-Fi connectivity, making it capable of connecting directly to Wi-Fi networks and cloud services. The Microchip board also has integrated Wi-Fi connectivity and can easily connect directly to Wi-Fi networks and cloud services. Unfortunately, the NXP board does not have integrated Wi-Fi and uses ethernet to connect. Once again this is not an absolute deal breaker since you can connect a Wi-Fi module through the expansions, although this adds even more cost to the project.

Our thermostat also needs a certain amount of Flash and RAM to operate correctly. Flash is non-volatile, used for long term storage, has slower access times, and limited write cycles. RAM is volatile, used for temporary storage, has faster access times, and unlimited write cycles. All of the board mentioned have enough of both types of memory and are probably overkill for our application with the TI board having 1MB Flash and 256KB RAM, the Microchip board having between 512KB and 2 MB Flash and between 128KB and 512KB RAM, and the NXP board having 1 MB Flash and 256KB RAM. As a matter of fact, NXP has a board that has all of the required peripherals and Wi-Fi but is a little short on the Flash and RAM requirements (FRDM Development Board for MCX A14x/A15x MCUs). If we could write the application to use less RAM and Flash, we could use this and it is significantly cheaper, but the extra Flash and RAM of the current boards allows us to add more functionality in the future.

Overall, it seems like the best choice would be the TI board for a few reasons. First it is the cheapest of the 3 boards. It has all the necessary peripherals for our thermostat so nothing else needs to be added. The last reason is we are already familiar with it, so it is the easiest to work with. One thing to consider would be to create our own board though. We can decrease the size and cost by using a more appropriate processor and only connecting the peripherals that we need. The development boards are great for creating a prototype, but in production we would want cost and size reduction.

Microchip Technology Inc. (n.d.). EV36W50A WFI32-IoT Board User's Guide. Retrieved from <https://ww1.microchip.com/downloads/aemDocuments/documents/WSG/ProductDocuments/UserGuides/EV36W50A-WFI32-IoT-Board-Users-Guide-DS50003262.pdf>

NXP Semiconductors. (2016). K64P144M120SF5 Data Sheet: Technical Data Rev. 7, 11/2016 Kinetis K64F Sub-Family Data Sheet. Retrieved from <https://www.nxp.com/docs/en/data-sheet/K64P144M120SF5.pdf>

Texas Instruments Incorporated. (2017). CC3220 SimpleLink™ Wi-Fi® and Internet of Things Technical Reference Manual. Retrieved from <https://www.ti.com/lit/ug/swru465/swru465.pdf>