Nicholas Glover

6/16/2023

**CS-350: Final Project Report**

The peripherals used by the thermostat created for this project are the UART, I2C, Timer, and GPIO peripherals. They deal with the various types of input and output that are necessary to make the thermostat work. The UART peripheral is used to make the thermostat output the state of the system to the terminal in order to simulate a connection to the cloud. The I2C peripheral is used to read information from the temperature sensor located on the board, so that the system can correctly interpret the information. The Timer peripheral is used to ensure that all the tasks necessary in the system are executed at the right times, so that the system runs smoothly and without error. Finally, the GPIO peripheral is used to handle button input so that the user can increase and decrease the temperature that the thermostat is set to.

The TI board used in the development of this thermostat is the CC3220SF-LAUNCHXL. It supports all of the necessary peripherals for the project and supports all the technical and physical requirements needed for the project. It supports the required peripherals, has the necessary onboard temperature sensor, and has 256 KB of RAM and 1 MB of executable flash memory, which is enough to handle the code for the project. It would be a great fit for further development and for connecting the thermostat to the cloud since it has a great many features for Wi-Fi connectivity and the security required for the creation of an IoT device like a smart thermostat. I was able to find a board from Microchip that has the necessary hardware to also be used to develop this project further. The WFI32-IoT Development Board can support the UART, GPIO, I2C, and Timer peripherals, it can be used to develop Wi-Fi solutions, and it has enough RAM and flash memory to handle the code. It has the necessary temperature sensor and would be able to connect the thermostat to the cloud. The only downsides to choosing this board are that it is more expensive than the TI board and it has fewer features overall. Finding a board from NXP (who bought Freescale) was very difficult because I could find any board that had all the necessary components to develop the project further. The closest I was able to come was with the Kinetis-series of microcontrollers. They can support the required peripherals (UART, I2C, GPIO, and Timer), but they don’t have an onboard temperature sensor or any Wi-Fi capability. Therefore, I don’t recommend the use of that chip for the future development of this project. Overall, I would recommend the TI board to further develop this project because it is an affordable, feature-rich board, that has all the necessary components to handle the future of this project.

References

Microchip Technology. (2022, February). WFI32-IoT Board User’s Guide. Microchip. <https://ww1.microchip.com/downloads/aemDocuments/documents/WSG/ProductDocuments/UserGuides/EV36W50A-WFI32-IoT-Board-Users-Guide-DS50003262.pdf>

NXP Semiconductors. (n.d.). K series cortex-M4. NXP. <https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/k-series-arm-cortex-m4:KINETIS_K_SERIES>

Texas Instruments. (2020, March). CC3220 SimpleLinkTM Wi-Fi® LaunchPadTM Development Kit Hardware. TI. <https://ti.com/lit/ug/swru463c/swru463c.pdf?ts=1686617856669>

Texas Instruments. (2021, May). CC3220R, CC3220S, and CC3220SF Simplelink Wi-Fi® single-chip wireless MCU Solutions. TI. <https://www.ti.com/lit/ds/symlink/cc3220sf.pdf>