**Architecture Report**

The thermostat prototype implements the required functionality using the four peripherals outlined below. Three types of wifi-enabled architectures have been put forward: TI (which is currently being used), Microchip, and Freescale. The goal of this report is to recommend the best one going forward based on their support for the necessary peripherals, Cloud connectivity, and their Flash and RAM capabilities.

**Peripherals**

I2C: enables the other peripherals to communicate, such as the thermometer with the UART.

GPIO: controls the buttons used to adjust the set temperature, and turns on or off the LED light based on whether the temperature is increasing.

UART: communicates to the server via Wi-fi, connecting the board to the console for displaying the output.

Timer: keeps the time for the time-controlled functions of the program.

All three proposed smart thermostat architectures are able to support all of the necessary peripherals.

**Cloud**

All three architectures can connect the thermostat prototype to Wi-fi, which means that they can use standard network protocols to communicate with Cloud servers.

**Flash and RAM**

All three architectures technically have sufficient Flash and RAM capabilities to support the thermostat program. However, Freescale’s one-time programmable (OTP) memory would make it impossible to re-program the thermostat, which is an undesirable limitation. Given this limitation, TI or Microchip would be better choices. I would choose the TI architecture, since the prototype is already using this and it works well.

# **References**

*CC3220SF SimpleLinkTM Wi-Fi® Single-Chip Wireless MCU Solutions*. CC3220SF datasheet | TI.com. (n.d.). https://www.ti.com/document-viewer/cc3220sf/datasheet

*Empowering Innovation*. (2024). Microchip.com. https://ww1.microchip.com/downloads/en/Appnotes/Atmel-42209-Thermostat-with-Touch-and-Wireless-Connectivity-Hardware-User

*Documentation*. NXP Semiconductors. (n.d.). https://www.nxp.com/design/documentation:DOCUMENTATION