**Task Scheduler Operational Documentation**

The prototype for the smart thermostat utilizes a TI CC3220SF MCU that has a built-in temperature sensor and SimpleLink Wi-Fi capabilities although the program can be modified to work on a different MCU as required. This microcontroller comes equipped with many useful features, of these the prototype utilized the built-in timer, LED indicator lights, two input buttons, the UART2 communication channels and the temperature sensor. After the software is loaded to the MCU it initializes a timer, every 100 milliseconds a local timer variable is incremented. If the user presses one, or both, of the buttons a flag is raised indicating a button was pressed. When the local timer variable is divisible by 2, indicating 200ms have passed, the task scheduler checks if one of the buttons has been pressed and adjusts the set time accordingly before resetting the button flag. The button presses add or subtract a degree from the set time based on which button the user has pressed. When the local timer variable is divisible by 5, indicating 500ms have passed, the task scheduler checks the ambient temperature via the I2C temperature sensor interface and compares it to the set temperature. If the ambient temperature is lower than desired the heater is turned on, and a red LED is lit to indicate the heater is set to the ON state. If the ambient temperature is greater than or equal to the set temperature the heater is set to the OFF state and the red LED is turned off. When the local timer variable reaches 10, indicating a full second has passed, the task scheduler sends the requested date to the server via the UART2 bidirectional communication channels and resets the local timer variable. The timer runs on a continuous call-back as long as the device remains powered. A state machine diagram is attached to provide a clear illustration of this functionality.

**Peripherals**

The thermostat prototype on the TI microcontroller supports the use of the peripherals: UART2 (Asynchronous bidirectional communication), I2C (integrated circuit interface for temperature sensor) and GPIO (general purpose input/output). By including the header files for these peripherals, it was relatively simple to initialize the peripherals and set the inputs and outputs with a quick and clean function calls. This made generating code that will be easy to follow and maintain. Inspecting the user guide for the Microchip MCU WFI32-IoT 2.0 Board and that MCU seems very similar in features and abilities. Both the Microchip and Freescale MCU supports sending to a cloud server although it uses the older UART instead of UART2. The microchip MCU uses a very similar temperature sensor with the I2C peripheral although its sensor range is -40°C – 125°C compared to Texas Instruments -40°C to 85°C. Either sensor range is sufficient for HVAC needs as homes typically are well within the ranges. Freescale has a development kit specifically for HVAC systems that uses the ZigBee software package, it is the 1321x-SRB kit. The information does not give an absolute range for the temperature sensor but does discuss LED lighting to indicate various ranges from -5°C to 40°C. The Zigbee package also has starter code for switching from Celsius to Fahrenheit as desired. All 3 microcontrollers have 2 buttons that can be programmed to set and adjust the desire temperature setting. They all have more than enough GPIO channels that to receive input and send data out. For this project regarding peripherals any of these 3 MCU’s can be utilized for the thermostat task scheduler. An additional consideration is Texas Instruments’ public forums, they help experts collaborate and provide answers for challenges, giving that device an edge for long term maintainability.

**WIFI Connectivity**

The Texas Instruments CC3220 MCU provides safe and secure WIFI connectivity using the SimpeLink Internet-on-a-chip. According to their documentation this chip offers “a wide range of built-in security features to help developers address a variety of security needs, which is achieved without any processing burden on the main microcontroller (MCU)” (Texas Instruments, 2021). It uses SSLv3, TLS1.0, TLS1.1, and TLS1.2) protocols with a trusted Root-certificate catalogue and Wi-Fi transmission power of 18.0 dBm at 1 DSSS or 14.5 dBm at 54 OFDM (Texas Instruments, 2021). The Microchip MCU also has a separate chip to manage connectivity and uses the “WFI32E03PC module, which has a built-in cryptographic device that helps to make a secure and authenticated cloud connection” (Microchip, 2024). Freescale’s MC1320x MCU also supports WIFI connectivity but the documentation I found online is outdated, 2006, and seems far more complicated than the newer processes used by TI or Microchips.

**Flash and RAM**

In the world of embedded systems flash memory refers to the nonvolatile memory that is retained if the device loses power and RAM is volatile but higher speed memory used to store data needed during execution. The Freescale MC13213 supports software with” 60KB of flash memory and 4KB of RAM” (Freescale, 2007). Texas Instruments CC3220SF uses 1MB of executable flash and 256KB of RAM” (Texas Instruments, 2017). Microchip’s MC1321 has “2MB flash memory and 640 KB RAM” (Microchips, 2024). For available memory this makes Microchip’s MC1321 the superior choice.

**Conclusion**

Both the Texas Instruments MCU and Microchip’s MCU would be sufficient moving forward with the project. I like Texas Instruments’ forum for expert collaboration but Mircochips’ MCU has better memory. Price may become the next biggest consideration as they both meet requirements to sustain the application.

**References**

Freescale Semiconductor. (2007). *13213 Evaluation Kits User Guide*. NXP. https://www.nxp.com/docs/en/user-guide/13213EVKUG.pdf

Mircrochip Technology Inc. (2024). *WFI32-IOT 2.0 Board User’s Guide*. Microchip. https://ww1.microchip.com/downloads/aemDocuments/documents/WSG/ProductDocuments/UserGuides/WFI32-IoT-2.0-Board-User-Guide-DS50003690.pdf

Texas Instruments Inc. (2021, May). CC3220R, CC3220S, and CC3220SF SimpleLinkTM Wi-Fi® Single-Chip Wireless MCU Solutions. Texas Instruments. Retrieve from <https://www.ti.com/lit/ds/symlink/cc3220sf.pdf?ts=1734283123919&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC3220SF%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Depd-con-null-44700045336317314_prodfolderdynamic-cpc-pf-google-ww_en_int%2526utm_content%253Dprodfolddynamic%2526ds_k%253DDYNAMIC+SEARCH+ADS%2526DCM%253Dyes%2526gad_source%253D1%2526gclid%253DCjwKCAiAmfq6BhAsEiwAX1jsZ3MtgeX2flVwsbYPquPR7RC2QAqe0vQvWhjE8diwJxB6vboUvU6_5xoCq88QAvD_BwE%2526gclsrc%253Daw.ds> on Dec. 14, 2024.

Texas Instruments Inc. (2017, February). *CC3220 SimpleLinkTM Wi-Fi® and Internet of Things*. Texas Instruments. https://www.ti.com/lit/ug/swru465/swru465.pdf