Creating a thermostat requires specific components: the peripherals, connecting to wi-fi/cloud, and the different forms of memory. The peripherals used to create the thermostat are the I2C, UART, GPIO, and the Timer.

The first of these is the UART. The UART is used for displaying the output of the code to the terminal. For the most part, UART, is commonly used for displaying information to other screens via wi-fi. For the case of this project, it is mainly simulated through a serial connection. The biggest part of understanding the UART is that “its supported protocols make it ideal for wireless communications such as Bluetooth®” (*Universal Asynchronous Receiver and Transmitter Peripherals | Microchip Technology*, n.d.).

The next peripheral, I2C, is for “allow[ing] multiple "peripheral" digital integrated circuits ("chips") to communicate with one or more "controller" chips” (*I2C - SparkFun Learn*, n.d.). The use of this is for reading the information the UART displays. The use of I2C can be found in the TI architecture through the temperature sensor foud on the board itself.

The next peripheral is GPIO. In is “a set of pins on your computer's mainboard or add-on card” (Butler, 2022). The idea is that this is for reading external device connections. In the code itself, the GPIO is called for registering information from the buttons on the side of the board and the lights. This is specific to how the thermostat works in increasing or decreasing the temperature.

The physical architecture comes from TI in the form of the CC3200 board. This board is capable of running all of these different peripherals when called upon. To accomplish this, the board itself has the temperature sensor on it, 256KB of ram, and 1MB of flash memory. As a part of this, the microcontroller helps for creating the thermostat code and running of it. This is evident in the fact that there is capability for controlling LEDs, buttons, the on-board sensor, and even connect to wi-fi. The last item, wi-fi, is especially important in being able to use the cloud. In terms of what can be seen on the Freescale end is that it appears to exist anymore. This is because it has become owned by NXP. There are components of the Freescale architecture, but mostly in portions of the buttons, LEDs, and wi-fi. The difference in this part is that all of these are missing components.

With the three architectures, TI, Microchip, and Freescale, the device is capable of connecting to the internet. From there, the microcontroller follows standard TCP/IP protocols. With these protocols being followed, it is more readily able to connect to the cloud and have those resources. As indicated earlier, the UART simulates this connection.

The last component is the flash and the RAM. All of the architectures discussed support these. The flash memory is where the code is permanently stored. The RAM only works when everything is executed. That’s why unplugging your board will cause it to restart from the beginning of the program. The different architectures allow for accessing both of these types of memory components. With the Freescale, it is working on the components for the code to access the information from the sensor and send it through the non-volatile Flash and then to the volatile RAM for current action. The microcontroller is what is accessing the flash originally and sending the information in, while the TI board is where everything is held to run the different peripheral information that the code located in the Flash holds. All in all, for a basic thermostat as was done in this project, this worked well.

*References/Citations*

Afterburner. (2024). *GPIO capability*. GPIO\_V3.4.6.pdf. http://www.mrjones.id.au/afterburner/assets/files/GPIO\_V3.4.6.pdf

Butler, S. (2022, April 11). What is GPIO, and what can you use it for? *How-To Geek*. https://www.howtogeek.com/787928/what-is-gpio/

*CC3200 data sheet, product information and support | TI.com*. (n.d.). https://www.ti.com/product/CC3200

*CloudConnectTM | Network Thermostat*. (n.d.). https://networkthermostat.com/software/cloudconnect

*EV36W50A*. (n.d.). https://www.microchip.com/en-us/development-tool/ev36w50a

*I2C - SparkFun Learn*. (n.d.). https://learn.sparkfun.com/tutorials/i2c/all

*NXP® Semiconductors official site*. (n.d.). https://www.nxp.com/

Texas Instruments, N. (2015). *CC3200 Simplelink Wi-Fi and internet-of-things solution, ...* CC3200 SimpleLinkTM Wi-Fi® and Internet-of-Things Solution, a Single-Chip Wireless MCU. https://www.ti.com/lit/ds/symlink/cc3200.pdf

*Universal Asynchronous Receiver and Transmitter Peripherals | Microchip Technology*. (n.d.). https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/core-independent-and-analog-peripherals/communication-connectivity-peripherals/uart-peripherals