For my Thermostat project I created code that initializes a timer that when set checks the buttons every 200ms, then checks the temp every 500ms, and updates the LED according to the set temperature all while reporting to the server every second. I used UART to create this loop, after initializing the driver using variable buttonCounter, using an if while function to continue to check the buttons every 200ms. Then I used a tempertureCounter variable for an if while function to check the temperature.

Next, I created a code that when you push a button it increases or decreases the temperature set-point by 1 degree every 200ms, while turning off the LED if it excessed the set-point temperature. I implemented a function that if the button is pressed move the degree by 1, and if it is greater than the set point of 1 then the LED will turn off. I used a float temperature variable to get the current temperature value.

To turn on or off an LED when a user types ON or OFF into the console I started by modifying the base code to read the incoming byte and if it is correct turn on the LED, or off. I would use the GPIO\_write() function along with the LED\_OFF; LED\_ON; and LED\_GPIO\_PIN configurations.

I created a task scheduler with a typedef which is a construct that creates a new data type. I made two new variables in the type struct called task BL\_task and task TL\_task, I was able to start the task to check every 500ms.

I was unable to create a finger sensor, I understand that I would need to set up the GPIO pins according to the Launchpad that would communicate for the sensor. To initialize the sensor I set up the I2C sensor. The loop would continuously go around until the finger sensor interrupts the loop. Then analyze the data to the LED.

The peripherals in this project must be supported for the thermostat to work. The thermostat directly using the TI Board to blink the LED if the temperature is greater than the set temp. The microchip is used to help communicate with the computer. A Freescale semiconductor are in microcontrollers ranging from 8-32 bit.

The microcontroller that has both the semiconductor and the microchip uses flash and ram. In fact, the size of the microcontroller depends on the size of the RAM, and Flash Memory. The TI Board served as a mini microcontroller holding both. RAM is faster of the two and is random access memory versus flash memory is primarily used for storage.

My TI Board never would do anything different than the yellow LED lights blink, not effected by my program. I did do research and tried to debug but was unsuccessful.

References

Vahid, F., Givargis, T., & Miller, B. (2015). *CS 350: Emerging Systems Architectures and Technologies*. Zyante, Inc.