In order to create a thermostat prototype for SysTex there were a number of requirements that must be met. When considering what microcontroller we should use, it is important that it supports GPIO in order to control the LED, UART to send data to a server, I2C to get a reading from the temperature sensor, it is also important that it supports Wi-Fi connectivity and has enough RAM for the final build. For my prototype, I used the TI CC3220S LAUNCHXL, which includes the requirements previously outlined as well as being capable of connecting wirelessly which ensures future-proofing as well.

The **TI** architecture works incredibly well for the thermostat project, as it typically provides integrated peripherals that are essential for a product such as a thermostat. These include I2C sensors that can be used for temperature, PWM modules that could be used for controlling an HVAC system, and UART interfaces which is crucial for communication with other components such as Wi-Fi modules. TI microcontrollers tend to be low-power, making them an ideal choice for a thermostat. **Microchip** microcontrollers offer many of the same peripherals that TI’s microcontrollers do, such as PWM and communication interfaces. However, microchip devices are known for their versatility, which would make them more suited for something like the HVAC system itself, rather than our use-case which is just to monitor the temperature. **Freescale** microcontrollers offer power cores combined with a range of peripherals. These microcontrollers provide a robust platform for the thermostat, but in our use-case they would more than likely be overkill. When it comes to RAM and Flash, TI microcontrollers offer a range of options, in the case of the CC3220S it includes 1MB of Flash memory and 256KB RAM. Looking at Microchip’s PIC16F18877, it includes up to 56 KB of Flash memory, 256B of EEPROM, and up to 4KB of SRAM memory. Freescale has now merged with NXP, so for them I chose to look at the FRDM-K64F which includes up to 1MB flash memory and 256 KB of RAM.

Regardless of which microcontroller is chosen, integrating Wi-Fi should be fairly simple if we utilize a compatible Wi-Fi module. This module would interface with the microcontroller via UART, SPI, or I2C and allow cloud connectivity through HTTP to transmit data securely to a cloud service such as AWS , Google Cloud, or Azure.