Caio Mauro

CS350

**THERMOSTAT**

In this project, the thermostat is designed to support peripherals using different hardware architectures including TI, Microchip, and Freescale.

**TI Architecture:**

* Peripheral Support: The TI architecture is supported using TI drivers created by Texas Instruments. This includes drivers for GPIO, Timer, UART, and I2C peripherals, which are needed for interfacing with buttons, LEDs, serial communication, and I2C temperature sensors respectively.
* I2C Configuration: The ti\_drivers\_config.h file configures the I2C driver for TI architecture.
* Initialization: The initUART and initI2C functions initialize UART and I2C drivers, enabling communication with external devices.
* Temperature Reading: The readTempSensor function reads temperature data from the sensor using I2C communication for thermostat functionality.
* Timer Support: The initTimer function initializes and starts a timer for periodic tasks, like updating temperature readings and controlling the thermostat.

**Microchip Architecture:**

- Microchip architecture would require similar driver configurations and initialization procedures as the TI architecture. Specific Microchip drivers and configurations would be utilized instead of TI's.

**Freescale Architecture:**

- Like the Microchip architecture, Freescale architecture would require specific drivers and configurations for Freescale's hardware.

**Wi-Fi Connectivity:**

- To connect the thermostat to the cloud via Wi-Fi, additional components such as Wi-Fi modules and corresponding drivers would be required. The implementation would involve configuring Wi-Fi communication protocols and establishing connections to cloud services using libraries or SDKs provided by the Wi-Fi module manufacturer.

**Flash and RAM Considerations:**

- Each architecture has its own Flash and RAM specifications, which impact code storage and execution. Typically, the code size and RAM usage would depend on the complexity of the application, the number of peripherals used, and the efficiency of the code. Analysis and profiling would be necessary to determine Flash and RAM requirements for each architecture.