Module 7

* **Explain how the thermostat supports the peripherals used in the project. Make sure that you have included all the required details from the scenario in your report.**

The thermostat system utilizes the GPIO, Timer, I2C, and UART peripherals. The GPIO is used to control an LED, turning it on or off depending on the temperature reading and the setpoint. GPIO pins are also configured as inputs with pull-up resistors and falling-edge interrupts to detect if any buttons are pressed. The callbacks “gpioButtonFxn0’ and ‘gpioButtonFxn1’ set flags to indicate which button was pressed by the user. A timer is also configured to generate periodic interrupts set at every 100 milliseconds. The ‘timerCallback’ function sets a flag, ‘TimerFlag’, to indicate that the timer interrupt has occurred. This flag is also utilized in the main loop to manage tasks. I2C is utilized to communicate with the temperature sensor, The ‘readTemp’ function performs I2C transactions to read the temp data from the sensor as raw data then converts it into Celsius. Finally, UART is used to send the formatted temperature, setpoint, heater status, and time data to the terminal for monitoring.

* Explain how the thermostat connects to the **cloud** via Wi-Fi.

TI’s CC3220SF microcontroller includes integrated wi-fi. The simplelink SDK provides APIs to connect to available wi-fi networks and cloud services. This begins by initializing the wi-fi module, connecting to the network using correct credentials, and then use MQTT or HTTP client libraries to send this data to the cloud. Microchip’s PIC32MZ microcontrollers can use external wi-fi modules. The harmony framework provides drivers and libraries to accomplish wi-fi connectivity. First, use the harmony configurator to set up the wi-fi module, connect to the network, then use the TCP/IP stack to communicate with cloud services. Finally, for Freescale, NXP’s Kinetis K64F microcontrollers can use external wi-fi modules such as an ESP8266. The MCUXpresso SDK provides support for wi-fi drivers and connectivity libraries. An example would entail configuring the wi-fi module using AT commands or libraries, connecting to the network, then utilizing the IwIP stack for cloud communication.

* Discuss the architecture’s **Flash and RAM** that supports the code.

Each architecture’s flash and RAM specs play a critical role in supporting the application. TI utilizes 1mb of on-chip flash with 256 KB of SRAM. This ample amount of flash memory can allow large application code and multiple library integrations while the SRAM ensures smooth operation and data handling. Freescale utilizes 1 MB of flash with 256 KB of SRAM. Freescale is similar to the TI microcontroller in which the config supports the storage of application code and task execution with adequate runtime memory. Microchip offers up to 2 MB of flash memory and up to 512 KB of SRAM. This large amount of flash memory allows large application codes and multiple library integrations. The extra SRAM ensures that data handling is smooth and efficient.

This thermostat system integrates GPIO, Timer, I2C, and UART to control and monitor the temperature. Each of the three architectures provide more than enough support for each of the peripherals in the project. Also, each offers solutions for connecting to the cloud with wi-fi. The flash and RAM capacities of each architecture are well-suited to support the applications code. All of this comes together so that the thermostat system can achieve cloud connectivity, reliable temp control, and peripheral management.

Sources

*ESE101: Microcontroller Peripherals, GPIOs, and Blinking Lights: Part 1*. (2016, May 17). Embedded. https://embedded.fm/blog/2016/5/16/ese101-peripherals-part-1

‌

*9 Essential Microcontroller Peripherals Explained – Embedded Inventor*. (n.d.). https://embeddedinventor.com/9-essential-microcontroller-peripherals-explained/

‌

*Execution from RAM in Embedded systems*. (2021, August 1). GeeksforGeeks. https://www.geeksforgeeks.org/execution-from-ram-in-embedded-systems/

‌

TI. (n.d.). *CC3220SF data sheet* (TI, Ed.) [Review of *CC3220SF data sheet*]. Www.TI.com. Retrieved June 18, 2024, from https://www.ti.com/product/CC3220SF?bm-verify=AAQAAAAJ\_\_\_\_\_\_qQ\_4VXzujnN5GByPmjlt2TkDs1DbPL-FdCJIQP26QgdMu-\_jPSRifzdnuHron6I0IwH7GXBf5Dp06fGGiDXqbpA8O4Mu8DLxohloB8nfkIGTnvQTI-NP8fuWW-6YoqYxhatZX-S2M5V3h\_Kgv3\_os6DLuPlcMRgWrh5zVWbA20TqvkWuRu8KvJbwVcwyHMGMBxrmuFXkitBTs9\_iPj8wBKZBTsLohiFdm\_1uPHPzYmSwkuPbq1lJU8eOlqeErxJ6i\_mAomYsDG-j0TbcoYmufU2e\_caxBPlKyZ9veadEfMISRKft46HCz-#tech-docs

‌

Microchip. (n.d.). (Microchip, Ed.). PIC32MZ-W1 and WFI32 Family of Wi-Fi® Microcontrollers. Retrieved June 18, 2024, from https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/32-bit-mcus/pic32-32-bit-mcus/pic32mz-w1

‌

*K64-120 MHz*. (n.d.). Www.nxp.com. Retrieved June 23, 2024, from https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/k-series-arm-cortex-m4/k6x-ethernet/kinetis-k64-120-mhz-256-kb-sram-microcontrollers-mcus-based-on-arm-cortex-m4-core:K64\_120

‌