**Task Scheduler Documentation**

**Overview and Algorithm Description**

Created for SysTec, this task scheduler was designed to manage the core operations of a smart thermostat prototype. This is accomplished by coordinating periodic tasks required for monitoring the temperature of a room, handling user inputs, and reporting its data. My algorithm involves using a 200ms timer interrupt to drive the execution of the various tasks at different intervals (200ms, 500ms, and 1 second). Software-based counters are put in place to ensure the device is reading temperatures, adjusting the set-point temperature via GPIO buttons, controlling the heater (LED), and transmitting its data efficiently and at the correct time. This approach allows for reliable thermostat functionality in combination with maintaining precise control and responsiveness.

**Inputs**

* **Temperature Sensor Data (I2C):** Interacts with and reads temperature data from the TMP006 sensor.
* **Button Presses (GPIO Interrupts):** Handles user input from two buttons (GPIO\_BUTTON\_0 and GPIO\_BUTTON\_1) that can adjust the set-point temperature. Can occur every 200ms.
* **Timer Interrupts (Every 200ms):** Acts as the main trigger for scheduling individual tasks.

**Outputs**

* **Heater Control (LED):** Representing activating a heater, it controls the device’s LED based on the comparison between the current temperature and the set-point temperature. A lower temperature than the set-point enables the heater. Every 500ms, the temperature is read and the heater is controlled.
* **UART Data Transmission:** Sends formatted data for the room temperature, set-point temperature, status of the heater, and elapsed time since the application has started.

**Expected Results**

* **Temperature Control:** The device should be able to accurately maintain the desired set-point temperature by turning the heater on or off.
* **User Input Responsiveness:** Button presses should result in immediate adjustments to the set-point temperature.
* **Data Reporting:** The UART should periodically send data updates in the proper format, being room temperature, set-point temperature, heater status, and elapsed time.

**How the Thermostat Supports Peripherals**

* **Temperature Sensor (I2C):** The TMP006 sensor is connected via the Inter-Integrated Circuit Interface (I2C) bus. The driver for the I2C is initialized during startup, and the room temperature is read every 500ms using the “readTemp” function. This data is then used to determine whether or not the heating system should be activated.
* **LED (GPIO):** Controlled using the General-Purpose Input/Output (GPIO) pins, this represents whether or not the current temperature is below or above the set-point and if the heater is active or not.
* **Buttons (GPIO Interrupts):** Handled through interrupt service routines (ISR), two buttons are configured to increase or decrease the set-point temperature.
* **UART Communication:** Used to simulate the transmission of data to the cloud, the UART sends formatted data every second that represents the current state of the thermostat.

**Hardware Architectures Comparison**

1. **Texas Instruments (TI) CC3220S-LAUNCH XL**

* **Peripheral Support:** TI’s SimpleLink CC3220S-LAUNCHXL supports a variety of peripherals natively, including I2C, GPIO, and UART. These peripherals are integrated with TI’s SDK, making them highly compatible with the requirements of this project. The SDK also provides ready-to-use libraries for handling sensor interfacing and communication protocols.
* **Wi-Fi Connectivity:** The CC3220S features a built-in Wi-Fi Internet-on-a-chip module that includes embedded TCP/IP and TLS/SSL stacks, an HTTP server, and various internet protocols (Texas Instruments, 2017). This enables seamless IoT integration and support for secure cloud connectivity.
* **Flash and RAM:** The CC3220SF comes with 1024KB of on-chip flash memory and 256KB of zero-wait state on-chip SRAM (Texas Instruments, 2017). This is sufficient for the thermostat’s code.

1. **Microchip PIC32MZ-W1**

* **Peripheral Support:** Microchip’s PIC32MZ-W1 supports a wide range of communication peripherals, including I2C, GPIO, UART, and SPI (Microchip, 2020).
* **Wi-Fi Connectivity:** The PIC32MZ-W1 family features built-in Wi-fi support with its on-chip IEEE 802.11b/g/n compliant Single Input Single Output (SISO) WLAN interface with integrated transceivers (Microchip, 2020).
* **Flash and RAM:** The device has 1MB of flash memory for programs, 64KB of flash for a boot program, and 256KB of SRAM for both programs and data (Microchip, 2020). Like with the CC3220S-LAUNCH XL, this can handle the project’s requirements.

1. **Freescale (NXP) Kinetis K61**

* **Peripheral Support:** The Kinetis K61 microcontroller has comprehensive peripheral support, including multiple I2C, UART, and GPIO interfaces (NXP Semiconductors, 2018).
* **Wi-Fi Connectivity:** Freescale’s microcontroller is an Ethernet device and, as such, does not have built-in Wi-Fi. Despite this, it can be paired with Wi-Fi partner modules offered by NXP (NXP, 2024).
* **Flash and RAM:** The Kinetis K61 series offers up to 1024KB of flash and 128KB of RAM, which, while lower than the other two microcontrollers, is still enough to support the thermostat’s code.

**References:**

NXP Semiconductors. (February 2018). K61 Sub-Family. NXP.com. Retrieved from: <https://www.nxp.com/docs/en/data-sheet/K61P256M150SF3.pdf>

NXP Semiconductors. (2024). Wi-Fi + Bluetooth + 802.15.4 Partner Modules. NXP.com. Retrieved from: <https://www.nxp.com/products/wireless-connectivity/wi-fi-plus-bluetooth-plus-802-15-4/wi-fi-plus-bluetooth-plus-802-15-4-partner-modules:WIFI-BLUETOOTH-MODULES>

Microchip Technology Inc. (2020). PIC32MZ W1 MCU and WFI32 Module with Wi-Fi® and  
Hardware-Based Security Accelerator Data Sheet. Microchip.com. Retrieved from: <https://ww1.microchip.com/downloads/aemDocuments/documents/WSG/ProductDocuments/DataSheets/PIC32MZ-W1-and-WFI32E01-Family-Data-Sheet-DS70005425.pdf>

Texas Instruments. (February 2017). CC3220 SimpleLink™ Wi-Fi® and Internet of  
Things Technical Reference Manual. TI.com. Retrieved from: <https://www.ti.com/tool/CC3220S-LAUNCHXL#tech-docs>