**Thermostat Project**

Joseph Veneski

Department of Computer Science, Southern New Hampshire University

CS-350 Emerging System Architecture and Technology

Professor Roland Morales

February 25, 2024

**Thermostat Project**

Import *gpiointerrupt* to IDE the same way as the last assignment: A screenshot of a computer

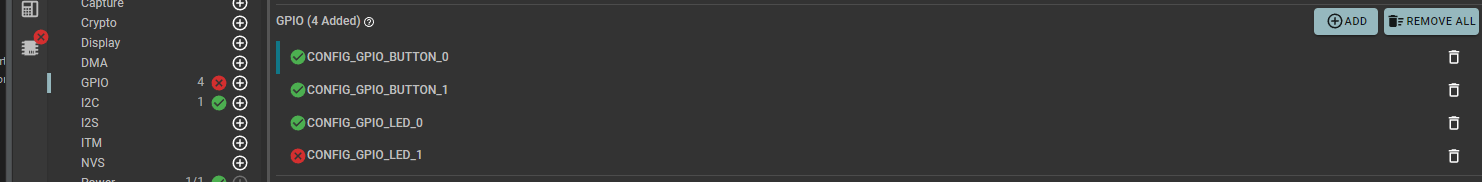
Description automatically generated

Build the project: A screen shot of a computer

Description automatically generated

Added Timer and I2C, resulting in conflict as described by the lab guide:A screenshot of a computer

Description automatically generated

Resolve the conflict by removing driver for GPIO LED 1: 

Added UART2 driver:

A screenshot of a computer

Description automatically generated

Delete code referencing GPIO LED 1 and rebuild without errors: A screen shot of a computer

Description automatically generated

Identify the correct COM Port: A close-up of a list of words

Description automatically generated

Successful output from program (ambient temp, setpoint, heat status, elapses seconds): A screenshot of a computer

Description automatically generated

State-machine diagram of Task Scheduler system (see attached pdf): A screenshot of a computer screen

Description automatically generated

**Algorithm:** The timer period is set to 100 milliseconds which is the greatest common denominator of the three task schedules. Every time the timer goes off, all three tasks are checked to see if it is their time to go off.

* Adjust setpoint temperature every 200 milliseconds
* Read/set ambient temperature every 500 milliseconds
* Set Heat based on ambient temp and setpoint every 1000 milliseconds

Inputs to the system include ambient temperature readings from the I2C sensor, as well as user input through GPIO buttons.

Outputs from the system include control of the Heat, status updates sent through UART, and adjustments to setpoint temperature based on user button input.

Expected results are whenever ambient temperature is below setpoint temperature, the heating element is turned on (LED ON) and when the ambient temperature rises above the setpoint (or equal) the heating element is turned off (LED OFF). Setpoint can be controlled through button input. Each timer cycle, all three functions are checked, updated if necessary and continue before the next time the timer goes off.

**Peripherals:** This project utilizes a range of peripherals including GPIO, I2C, UART2, and a Timer. The General-Purpose Input/Output is used to interface with the two buttons that the user can press to change the setpoint temperature. GPIO is also used to control the LED which is simulating the signal to enable the heat. Inter-Integrated Circuit is used to communicate with the microchip’s temperature sensor and record the ambient temperature. Universal Asynchronous Receiver/Transmitter is utilized to send status updates, messages, and errors to the terminal (simulated server). Finally, a Timer is utilized along with a task scheduler system to ensure tasks are done at consistent intervals.

Which microcontroller hardware architecture to choose depends on the needs of the project being worked on. Projects with emphasis on wireless connectivity solutions with low power requirements may want to investigate Texas Instruments (TI) boards such as the one used in this project. Microchip has extensive documentation and community support making it a good choice for beginners and hobbyists. NXP (previously Freescale) has a huge portion in the automative industry, had a leading role in developing near field communications and boasts a sizeable chunk of revenue from IoT sectors (Bylund, 2019). All three manufacturers support basic peripherals used in this project, although TI and Microchip offer the most extensive documentation for new users.

**Cloud:** The thermostat currently does not connect to the cloud; however this can be enabled by configuring credentials, a secure connection established with the cloud service provider using encryption, then transmitting data via HTTP or MQTT. Users would then be able to remotely control the thermostat via a device connected to the thermostat by the Cloud and the device could receive firmware updates. All three of the microcontroller manufacturers considered offer wireless connectivity solutions as required, however Texas Instruments has a reputation for wireless solutions.

**Flash & RAM:** Flash memory will be used for storing static data and program code. This includes the code we have written along with wi-fi connectivity and SSL/TLS communications and HTTP/MQTT logic. RAM will be used for temporary storage, variables, and the stack heap. There may be a small overhead from the basic operation as defined so far in the thermostat, plus additional overhead from the wireless connections. The board used for this project, TI’s SimpleLink CC3220S features 256KB of Flash, and 1MB of RAM (*CC3220S,* 2024). It is relatively easy to find boards with these specifications or higher by both TI and Microchip, however NXP which has a bigger focus on automotive use-cases may be more difficult.

In conclusion, it appears that the Texas Instruments SimpleLink CC3220S board that was utilized for this project remains an ideal choice for a wireless connected thermostat. Although all three manufacturers could be utilized for this type of project, NXP does not offer many boards meant for this type of work. Both Texas Instruments and Microchip have a fair amount of documentation facilitating development. Texas Instruments has a long history and excels at wireless connectivity in low-power solutions (*Wireless Connectivity*, 2024).

**References**

Byland, A. (2019). *Better Buy: NXP Semiconductor VS. Texas Instruments*. Yahoo Finance. <https://finance.yahoo.com/news/better-buy-nxp-semiconductors-vs-150700679.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAALBl2AhytV8-2EFoXTCTLn9wlj0yFuQrM3NlDAK-I4ZwLzXc6hf7TRa5sZElL9wlNUl26AwCEHTM9auLSgGP9FMsi8Y91RU2iupm_IzluL-2Wyn_3qlQqW8sw6ySnMmbtM5Q3wlVFKonCHkZv15z9kAwCp-J22Ka4Wu2x8iK9so4>

*CC3220S*. (2021). Texas Instruments. <https://www.ti.com/product/CC3220S>

*Wireless Connectivity*. (2024). Texas Instruments. <https://www.ti.com/wireless-connectivity/overview.html>