Final Project Report

Ryan Rieth

Prof. Trajkovski — CS 350 — SNHU

This final project is an amalgamation of every concept learned throughout CS 350 ranging from GPIO LED adjustment, timer utilization, UART console connection, I2C temperature sensing, and general state machine and TI board configuration. The project is purposed to highlight peripheral support alongside Cloud connection via Wi-Fi. Texas Instruments’ CC3220S board is utilized to render this project and all previous milestones throughout the course a reality.

The SimpleLink CC3220S TI board is an impressively powerful pocket-sized machine that allows a vast array of Cloud-related and code-cemented executions for students to properly learn how embedded systems can operate in the “real world.” Its different modes, support for use with multiple peripherals, and plethora of respectable features make Texas Instrument’s board a wonderfully capable powerhouse. It boasts up to 256KB of on-chip, zero wait state SRAM as well as 1024KB on-chip flash memory, allowing application code to execute while freeing up SRAM for the purposes of reading and writing data (*Texas Instruments*). The TI board also features Wi-Fi support and connectivity, tactile buttons, a temperature sensor, and LED lights that can be mapped to buttons and controlled manually or via scripted code.

As one can imagine, more than one company produce development boards that can accomplish the same tasks. Texas Instrument’s CC3220S board is one of many. Another capable device is Microchip’s AVR-IoT WA development board. This board also boasts Wi-Fi connectivity as well as four programmable LEDs, two mechanical buttons, a light sensor, a temperature sensor, and an on-board debugger (*Microchip*). Microchip’s board is equally priced as Texas Instrument’s — around $50 — and far below the substantial $499 cost of Freescale’s foray into an embedded systems development board: the NXP Cortex-AI i.MX6 ARM Android/Linux development board. Freescale’s board adds audio input/output functionality, such as a buzzer, as well as a substantial increase in GPIOs, COM ports, on-board memory (1GB), and even a keypad for an additional cost (*Geekland*). Beyond increases in specs, the Freescale board also boasts an SD card slot, ethernet, standard HDMI, and USB plug-ins that both Microchip and TI’s boards do not have.

Where Texas Instrument’s board succeeds, however, is in is invaluable combination of low cost, feature-rich capability, and ease of use. Combining the TI board with the Spyder IDE, scripting and control of the CC3220S board were beyond manageable. This final project required the use of a few peripherals: a timer, LEDs, buttons, a temperature sensor, and output to a serial terminal on the computer. With the project’s *sysconfig* file, the board’s configurations for each of these peripherals were easily adjustable to allow not only support of each peripheral but also dependent communication or execution between each peripheral based upon user-specified data. For example, the terminal outputs data relevant to the current temperature, a desired set point, whether the heater is turned on or off, and the elapsed time. A press of a button either increases or decreases the desired set point, and the temperature sensor on the board can be manually adjusted via a long finger press to increase the current temperature. Should this current temperature fall below the set point, the heater will turn on also triggering an LED light to indicate the heater is on. When the temperature reaches or passes the set point, the heater will turn off and this indication will reflect in the LED turning off. For the purposes of this project, manual temperature readings were taken and lights and buttons on the board itself were adjusted, but the Wi-Fi module can allow the board to communicate with wireless buttons or sensors spread throughout a user’s home to reflect variables in the scripted code very similarly. Instead of the LED light on the board turning on, for example, a buzzer device connected to the board’s Wi-Fi via IoT connectivity in the user’s home can ding whenever the temperature falls below the set point.

Functionalities as such are capable because of the board’s on-chip SRAM and flash memory. The two kinds of memory, in a sense, share the workload. Where the SRAM can continue reading and writing data to and from the device, the flash memory can execute the specified functions as readings are sensed or data is written. This two-hand approach maximizes efficiency within the board.

For the purposes of this project, and the goals of the Company reflected, Texas Instrument’s CC3220S board is more than capable of accomplishing the tasks at hand with the benefit of the user in mind. Its cost is manageable and quite a bargain for the rich list of features the board possesses. A respectable level of SRAM and flash memory allows proper and efficient execution of most functions; LED lights and clickable buttons allow manual intervention and visualization for functions and their results; a temperature sensor can monitor the surrounding environment for desired and even safety-critical levels; and overall Wi-Fi connectivity introduces the board’s vast feature set to the IoT and a massive world of other devices and functionalities, bringing control to a user’s smart home in an easy to use, understand, and maintain device.

## References

*Technical Reference Manual.* Texas Instruments. (2017)*.* CC3220 SimpleLink Wi-Fi and Internet of Things. Retrieved from <https://www.ti.com/lit/ug/swru465/swru465.pdf?ts=1686938220848>

*EV15R70A AVR-IOT WA Development Board*. Microchip. (n.d.). <https://www.microchip.com/en-us/development-tool/EV15R70A>

*NXP Cortex-A9 i.MX6 ARM Android/Linux Development Board*. Geekland. Industrial ARM Single Board Computer. (n.d.). <https://www.geekland.co/NXP-Cortex-A9-iMX6-ARM-Android-Linux-Development-Board-GK-IMX6D-V12.htm>