Gracie Ream

CS 350

4/18/2025

7-2 Final Project: Thermostat Lab Report/Reflection

Overall, for the final project, the thermostat successfully supported all the peripherals used. The peripherals being the temperature sensor, LEDs, LCD display, UART connection, and buttons. The buttons were to adjust the temperatures as well as changing the display and program to either off, heat or cool. The temperatures were shown on the display and changed when the buttons were pressed, the LEDs lit up and changed to either blinking or solid depending on the temperature and set temperature, and the UART serial connection allowed the data transmission between the script and the display.

Comparing the three different hardware architectures: Raspberry Pi, Microchip, and Freescale, all have pros and cons to the specified requirements of supporting peripherals, connecting to the cloud via Wi-Fi, and having enough Flash and RAM to support the code. Raspberry Pi can clearly support the necessary requirements as stated above and from seen with the project. It supports the buttons, LEDs, display, UART connection and the temperature sensor. Raspberry Pi also has built in Wi-Fi or can be connected as well as enough RAM and Flash storage to support this. Microchips can support their peripherals through MCU registers and libraries while having both Flash and RAM storage, though limited. Microchip also does not support Wi-Fi unless externally connected. Looking at Freescale, it can also support the required peripherals through SDK, but like Microchip, its Wi-Fi must be external. Freescale’s RAM and Flash Storage would be compatible with the required code.

All three architectures connect to the cloud via Wi-Fi differently. Raspberry Pi has built in Wi-Fi modules and runs Linux systems; it essentially connects to Wi-Fi like a laptop would. Using a programming language, the data can be sent to the cloud via HTTP protocol. Microchip microcontrollers do not have the same built in Wi-Fi as Raspberry Pi and must introduce an external module. It communicates with the Wi-Fi via UART or SPI and then connects to Wi-Fi. It also sends information to the cloud through a program language/something sent to the modules. Freescale also does not have built in Wi-Fi like Raspberry Pi and must be connected to an external module. Once connected to Wi-Fi, Freescale can communicate with the cloud through HTTP requests.

For supporting the code, Raspberry Pi can run programming languages like Python, C++ and C. It provides UART connection, gpiozero (for the buttons and LEDs), and can run a great level of code since it is uses Linux. Microchip usually runs a lower level of code than Raspberry Pi can (C/C++). However, it can interact with GPIO and UART as well. Freescale also is able to work with UART connection, GPIO control, and temperature control which are all necessary for this project.

**References**

Ostapiuk, R. (2023). 16-bit PIC® MCU Architecture - Developer Help. Microchip.com. <https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/16bit-mcu/architecture/>

Architecture of Raspberry Pi. (2023, September 1). GeeksforGeeks. <https://www.geeksforgeeks.org/architecture-of-raspberry-pi/>