CS350 Final Project Report

Tristan Maloy

10/19/2025

**Peripherals**

The thermostat integrates multiple peripherals including a temperature and humidity sensor (AHT20), LEDs, buttons, and a 16x2 LCD display. The Raspberry Pi functions as a fully embedded Linux platform that can provide access to GPIO, I2C, PWM, and UART. The buttons are connected to GPIO pins 12, 24, and 25. When these are pressed, they trigger specific state machine actions such as changing the mode of the thermostat and adjusting the temperature. The temperature and humidity sensor is connected through an I2C bus that helps to provide real time communication via the sensor. The LEDs are connected to GPIO pins 18 and 23 and they use PWM to fade in and out which helps to visually indicate whether the system is currently heating or cooling. The UART helps to communicate data across the server to update information in real time such as the LCD display without having to be used in the main control loop.

Microchip architecture is designed for embedded control. A microchip such as the PIC18-Q71 can easily configure peripherals while also providing various package and memory options to scale for different things such as LED lighting (with 3 16-bit dual PWMs), automation, and IoT. It also has a flexible serial communications interface that includes a UART that also supports I2C (Microchip). Freescale architecture, such as the Kinetis K-64, similarly supports the peripherals and are commonly used in HVAC systems because of their low power processing efficiency and their scalability for large memory (NXP, 2025).

**Cloud**

The Raspberry Pi uses Linux to manage Wi-Fi, therefore there is a reliance on the operating system when wanting to communicate and store data on the cloud. The PIC18-Q71 requires an external Wi-Fi module through a serial port like UART which can be utilized to support the lower power consumption that comes with the microchip. The Kinetis K-64 has an IEEE 1588 Ethernet MAC that will allow wired connection while also allowing UART support for an added Wi-Fi module. The Raspberry Pi using Linux provides less complexity of code as it doesn’t need the Wi-Fi to be integrated into the code, whereas the microchip and Freescale architectures would require there to be configuration and control through the software and/or firmware.

**Architectures’ Capabilities**

Each architecture supports the thermostat’s state machine logic, display management, and sensor communication, however each one comes with different levels of control, efficiency, and power usage. Using the Raspberry Pi would be ideal in situations where the device is being prototyped as it can use pythons built in drivers such as gpiozero and adafruit to help connect to the peripherals while also allowing native Wi-Fi connection and the ability to multitask real time while keeping functions outside of the main loop. Microchip and Freescale architectures are both more power efficient than the Raspberry Pi and require external modules to access Wi-Fi to allow cloud support. The microchip also contains a configurable CIP that increases system responsiveness while reducing power consumption which increases the microchips efficiency (Microchip). Freescale architecture provides independent flash banks that allow the execution of code and firmware updating that does not affect the performance of the Freescale (NPX, 2025).

**Resources**

Microchip. (n.d.-b). PIC18-Q71 microcontrollers (mcus) | microchip technology. https://www.microchip.com/en-us/products/microcontrollers/8-bit-mcus/pic-mcus/pic18-q71

NPX. (2025, January 16). *K64-120 mhz*. Arm Cortex-M4|Kinetis K64 120 MHz 32-bit MCUs | NXP Semiconductors. https://www.nxp.com/products/K64\_120