Ethan Mills

CS-350

Southern New Hampshire University

April 27, 2025

Final Project Reflection

In this project, the Raspberry Pi is the central controller in the thermostat system. It supports various peripherals, including GPIO pins and UART interfaces. In this case, the buttons are used to change the thermostat's state (heat, cool, or off). The LEDs are controlled using PWM to simulate fading in and out, indicating whether the system is heating or cooling. The temperature sensor is also connected to the device, allowing the Raspberry Pi to read the temperature and humidity that the sensor detects. It uses those readings to compare with the set point and control the system's state (to heat or cool). The LCD display is what provides the user feedback on the device's functions and displays results. In this case, it displays the current date and time, the temperature, and the set point.

The Raspberry Pi architecture is a multi-core processor with large computational power. This allows the device to support and manage multiple peripherals, including buttons, LEDs, temperature sensors, and the LCD display, all of which were utilized in this project. It also allows for the use of software libraries, such as Python libraries, to code the device and the functions of its peripherals. This device also has Wi-Fi capabilities. It features built-in Wi-Fi support, enabling easy connection to cloud services. The peripherals for this device are the GPIO pins, PWMS, Temperature Sensor, and LCD Display. The Raspberry Pi’s Wi-Fi capabilities enable it to connect to cloud services for remote control and monitoring. This allows for the cloud to be used to log data, control the system,m and apply system updates if needed.

Peripherals In Depth:

* Buttons: Control the State of the Machine, in this case, control the state of the Thermostat, Heat, Cool, and off, and the set point of the temperature.
* LEDS: The LEDs indicate the thermostat's current state; a fading red LED indicates the system is heating, and a fading blue LED indicates the system is cooling.
* Temperature Sensor: This sensor continuously measures the temperature and humidity of the room in which it is located. This data is then used to determine the state of the machine, if the system needs to cool or heat.
* LCD Display: The LCD display provides the user with feedback on the temperature, machine state, set point, and current date and time.

The Microchip architecture features low-level control microcontrollers designed to exert direct control over hardware, allowing them to manipulate registers with precision and fine-tune the operation of added peripherals. Microchip controllers are also more power-efficient, making them the ideal choice for battery-powered or lower-powered applications. Lastly, Microchip microcontrollers are more suited for real-time operations, making them excellent choices for systems that do not require precise timing or where low latency is crucial.

Freescale microcontrollers are optimized for embedded control systems, enabling them to run real-time tasks effectively and efficiently. However, they often require an external module, such as a Wi-Fi chip, to enable Wi-Fi connectivity. Much like Microchip, Freescale's architecture of microcontrollers is well-suited for low-power applications, such as embedded systems. Freescale’s scalability is also excellent, as it offers a range of models that support Wi-Fi and Ethernet capabilities. They are flexible and capable of scaling to fit the application in use.

For this project, this system uses a state machine to manage the thermostat's modes: off, heat, and cool. As mentioned above it uses the LCD display to provide the user feedback on the device's functions, displays results, current date and time, the temperature, and the set point. Depending on the current temperature the sensor is reading, the state machine controls the system's operation, turning it off, turning the heat on, or turning the cooling on. In this project, this system also utilizes UART communication to simulate sending data to a server. It sends information such as the current state, temperature, and set point every 30 seconds. This simulates what it would be like if the data were being sent to a cloud service or a server.

Work Cited

Brown, M. (2013, February 18). Freescale’s tiny ARM chip will put the Internet of Things inside your body. Wired. <https://www.wired.com/2013/02/freescales-tiny-arm-chip/>

GeeksforGeeks. (2021, April 30). Architecture of Raspberry Pi. <https://www.geeksforgeeks.org/architecture-of-raspberry-pi/>

Lee, J. (2017, June 5). Raspberry Pi developer's guide: GPIO. Junye’s Blog. <https://junyelee.blogspot.com/2017/06/raspbery-pi-developers-guide-gpio.html>

SparkFun. (n.d.). Introduction to the Raspberry Pi GPIO and physical computing. SparkFun Electronics. <https://learn.sparkfun.com/tutorials/introduction-to-the-raspberry-pi-gpio-and-physical-computing/gpio-pins-overview>

Wikipedia contributors. (2023, October 6). NXP ColdFire. Wikipedia. <https://en.wikipedia.org/wiki/NXP_ColdFire>

Wikipedia contributors. (2024, March 18). I.MX. Wikipedia. <https://en.wikipedia.org/wiki/I.MX>