Smart Thermostat Architecture Report

SysTec is planning to enter the smart thermostat market, which is expected to grow to nearly $9 billion by 2026. I was asked to build a low-level prototype of the thermostat using a development board. This report explains how I created the working prototype and explores the next step in the project, which is connecting the thermostat to the cloud. I also compare three different hardware architectures Raspberry Pi, Microchip, and Freescale to help decide which one is the best fit for the final product.

For the prototype, I used a Raspberry Pi board and created a thermostat that can detect temperature, show information on an LCD screen, and change heating or cooling modes with the press of a button. It also sends status updates through a serial connection. The prototype uses an AHT20 temperature sensor, three push buttons, two LEDs, and a 16x2 LCD. The Raspberry Pi supports all of these peripherals using its GPIO pins, I2C communication for the sensor, and serial communication for data output.

The next step is to connect the thermostat to the cloud using Wi-Fi. This feature will allow SysTec's server software to receive data and monitor the thermostat remotely. The Raspberry Pi has built-in Wi-Fi, which makes it easy to connect to the cloud. Microchip boards like the PIC32 and SAMD21 often need external Wi-Fi modules, which adds to the cost and complexity. Some newer Microchip boards include Wi-Fi but may have limited memory. Freescale boards like the i.MX series also offer built-in Wi-Fi and are used in real-world smart devices, making them a strong option for cloud connectivity.

To run the code from the prototype, the thermostat must have enough Flash memory and RAM. The Raspberry Pi comes with enough memory and storage to run Python code and multitask easily. However, it is not always reliable for commercial products and can be more power-hungry. Microchip boards use less power and are reliable for embedded use, but they usually have less memory, and the code needs to be rewritten in C or C++, which takes more time. Freescale boards have a good balance of memory and performance. They can run more complex software and support multitasking, which is useful as the thermostat grows more advanced.

Based on this analysis, I recommend using a Freescale board for the production version of the thermostat. It supports all of the required peripherals, has built-in Wi-Fi for cloud communication, and offers enough memory and processing power to handle the thermostat’s software. Raspberry Pi is great for prototyping, but not ideal for final products. Microchip boards are efficient but may require more development time and added hardware. Freescale provides the best mix of features and reliability for SysTec’s needs.