The global smart thermostat market is forecast to reach nearly $9 billion by 2026 (*Smart Thermostat Market Size, Share & Trends Analysis Report by Technology (LORAWAN, Wi Fi), by Product (Connected, Learning), by Installation (New Installation, Retrofit), by End Use (Residential, Commercial), by Region, and Segment Forecasts, 2025 - 2030*, n.d.) SysTec is looking to enter the smart thermostat arena, so I built a prototype on a Raspberry Pi. It reads room temperature, controls LEDs for heating or cooling, lets you change the temperature set point with buttons, shows status on a 16×2 LCD, and sends data over serial every 30 seconds.

The thermostat has three states: Off, Heat, and Cool. Press the mode button to go Off 🡪 Heat 🡪Cool🡪 Off.

* In Heat, the red LED blinks if the room is colder than the set point and stays solid once it reaches or exceeds that point.
* In Cool, the blue LED blinks if the room is warmer than the set point and stays solid once it reaches or goes below that point.
* Leaving Heat or Cool turns both LEDs off.

**Code Structure**

I organized the code into small functions and clear classes. Each part, sensor reading, light control, button handling, display updates, and serial output, lives in its own module. I added basic checks so a bad sensor read or write failure doesn’t crash the program.

**Peripherals**

* Temperature sensor on the I²C bus, sampled once a second.
* Two LEDs for heat and cool indicators.
* Three buttons: mode, increase, and decrease.
* A 16×2 character display updated every second.
* A serial connection sending “mode,temperature,target” every 30 seconds.

**Hardware Comparison**

**Raspberry Pi**

* Peripherals: onboard I2C, GPIO, PWM, and UART headers handle the sensor, LEDs, buttons, and display without extra hardware.
* Wi-Fi: built-in module connects directly to the network with standard socket or MQTT libraries.
* Memory: SD card storage and 1 GB RAM (Raspberry Pi, n.d.), more than enough for the thermostat firmware.

**Microcontroller Board**

* Peripherals: native I2C, GPIO, PWM, and UART, no additional interface chips required.
* Wi-Fi: external Wi-Fi module (e.g. WINC1500) using a lightweight TCP/IP stack.
* Memory: around 512 KB flash and 128 KB RAM, sufficient for our code and networking.

**NXP K64**

* Peripherals: integrated I2C, GPIO, PWM timers, UART—covers all low-level needs.
* Wi-Fi: external shield or module with MCUXpresso networking libraries.
* Memory: 1 MB flash and 256 KB RAM—ample room for code and optional RTOS.

**Recommendation**

I recommend a microcontroller board with an external Wi‑Fi module. It offers the right mix of size, power consumption, cost, and enough memory to run our thermostat code and handle cloud updates. The Raspberry Pi prototype meets all low‑level requirements. For production, the Microchip platform is the right choice to bring SysTec’s thermostat to market.

References:

Raspberry Pi. (n.d.). *Raspberry Pi Documentation - Raspberry Pi Hardware*. www.raspberrypi.com. <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>

*Smart Thermostat Market Size, Share & Trends Analysis Report By technology (LORAWAN, Wi Fi), by product (Connected, Learning), by installation (New installation, Retrofit), by end use (Residential, commercial), by region, and segment Forecasts, 2025 - 2030*. (n.d.). https://www.grandviewresearch.com/industry-analysis/smart-thermostat-market

‌