# **SysTec Smart Thermostat Prototype & Hardware Recommendation Report**

**Prepared by:** Jaden Williams  
**Role:** Embedded Systems Developer  
**Date:** 6/13/25

## **Executive Summary**

The prototype smart thermostat developed for SysTec successfully meets the initial functionality goals outlined by the CEO. The prototype demonstrates accurate temperature sensing, real-time status updates via LCD, interactive controls, visual heating/cooling indications, and serial data output—all designed to mimic a smart thermostat’s essential behavior.

This report details the architecture and software implementation of the prototype and provides a technical evaluation of suitable microcontroller architectures (Raspberry Pi, Microchip, and Freescale/NXP) to guide the next phase: integrating cloud connectivity.

## **Prototype Overview**

### **Core Features Implemented:**

* **Default Set Point:** 72°F
* **Temperature Sensor:** AHT20 over I2C
* **LED Indicators:** Red and Blue PWM LEDs for heating/cooling state
* **Control Inputs:**
  + Button 1: Cycles thermostat mode (Off → Heat → Cool)
  + Button 2: Increases set point
  + Button 3: Decreases set point
* **LCD Display:**
  + Line 1: Date and current time
  + Line 2: Alternates between current temperature and system state
* **Serial Output:**
  + Every 30 seconds: state,temp,setpoint (e.g., heat,71.8,72)
* **State Machine:** Used to manage thermostat logic and transitions

This phase used a Raspberry Pi with GPIO-controlled peripherals, running a Python-based state machine to simulate real-world behavior.

## **Cloud Integration Objectives**

The next development phase will focus on transmitting thermostat data (temperature, state, set point) to SysTec’s server over Wi-Fi. To support this, the production system must:

1. Support I2C for temperature sensors and GPIO for LEDs/buttons.
2. Support character LCDs or alternative display modules.
3. Have integrated or easily added Wi-Fi capability.
4. Provide sufficient memory for embedded logic and connectivity stacks.

## **Hardware Architecture Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature / Platform** | **Raspberry Pi (Zero W / 4)** | **Microchip (SAM D21 or SAME54)** | **Freescale/NXP (i.MX RT or Kinetis K64)** |
| **Wi-Fi Connectivity** | Built-in on Zero W & 4 | External Wi-Fi module needed | Some models include Wi-Fi, others need add-on |
| **Peripheral Support** | Full (I2C, GPIO, UART, PWM) | Full (I2C, GPIO, UART, PWM) | Full (I2C, GPIO, UART, PWM) |
| **Flash/RAM** | 512MB–4GB RAM, 256MB+ Flash | 256KB–1MB Flash, 32–256KB RAM | 512KB–2MB Flash, 256–1MB RAM |
| **Power Consumption** | High | Low | Low–Medium |
| **Programming** | Python/Linux-friendly | C/C++, RTOS or Bare Metal | C/C++, RTOS or Bare Metal |
| **Scalability** | High (Linux-based) | Moderate | High (many product lines) |
| **Cost** | $10–$55 | $2–$15 | $5–$20 |

## **Recommendation**

After evaluating the platforms, **the Microchip SAME54** is recommended for the production design. Here’s why:

### **Advantages:**

* **Peripheral Compatibility:** Supports I2C, PWM, and GPIO natively for sensors, LEDs, and buttons.
* **Low Power:** Better suited for long-term deployment of home automation.
* **Flash/RAM Sufficiency:** 1MB Flash and 256KB RAM are ample for the current thermostat firmware and networking stacks.
* **Wi-Fi Expandability:** Pairs well with Microchip's WINC1500 or ATWINC3400 Wi-Fi modules, which integrate seamlessly.

### **Raspberry Pi Considerations:**

While great at prototyping (especially with Python and existing Linux tools), it is **overpowered**, draws more current, and introduces complexity not required in a production thermostat.

### **Freescale/NXP Considerations:**

Powerful and scalable, but higher complexity and cost compared to Microchip solutions for the scope of this product.