Temperature Control Using a Microcontroller: 
An Interdisciplinary Undergraduate Engineering Design Project

James S. McDonald 
Department of Engineering Science 
Trinity University 
San Antonio, TX 78212

Abstract. This paper describes an interdisciplinary design project which was done under the author’s supervision by a group of four senior students in the Department of Engineering Science at Trinity University. The objective of the project was to develop a temperature control system for an air-filled chamber. The system was to allow entry of a desired chamber temperature in a prescribed range and to exhibit overshoot and steady-state temperature error of less than 1 degree Kelvin in the actual chamber temperature step response. The details of the design developed by this group of students, based on a Motorola MC68HC05 family microcontroller, are described. The pedagogical value of the problem is also discussed through a description of some of the key steps in the design process. It is shown that the solution requires broad knowledge drawn from several engineering disciplines including electrical, mechanical, and control systems engineering.

1 Introduction

The design project which is the subject of this paper originated from a real-world application. A prototype of a microscope slide dryer had been developed around an Omega™ model CN-390 temperature controller, and the objective was to develop a custom temperature control system to replace the Omega system. The motivation was that a custom controller targeted specifically for the application should be able to achieve the same functionality at a much lower cost, as the Omega system is unnecessarily versatile and equipped to handle a wide variety of applications.

The mechanical layout of the slide dryer prototype is shown in Figure 1. The main element of the dryer is a large, insulated, air-filled chamber in which microscope slides, each with a tissue sample encased in paraffin, can be set on caddies. In order that the paraffin maintain the proper consistency, the temperature in the slide chamber must be maintained at a desired (constant) temperature. A second chamber (the electronics enclosure) houses a resistive heater and the temperature controller, and a fan mounted on the end of the dryer blows air across the heater, carrying heat into the slide chamber.

This design project was carried out during academic year 1996–97 by four students under the author’s supervision as a Senior Design project in the Department of Engineering Science at Trinity University. The purpose of this paper is to describe the problem and the students’ solution in some detail, and to discuss some of the pedagogical opportunities offered by an interdisciplinary design project of this type. The students’ own report was presented at the 1997 National Conference on Undergraduate Research [1].

Section 2 gives a more detailed statement of the problem, including performance specifications, and Section 3 describes the students’ design. Section 4 makes up the bulk of the paper, and discusses in some detail several aspects of the design process which offer unique pedagogical opportunities. Finally, Section 5 offers some conclusions.

2 Problem Statement

The basic idea of the project is to replace the relevant parts of the functionality of an Omega CN-390 temperature controller using a custom-designed system. The application dictates that temperature settings are usually kept constant for long periods of time, but it’s nonetheless important that step changes be tracked in a “reasonable” manner. Thus the main requirements boil down to

- allowing a chamber temperature set-point to be entered,
- displaying both set-point and actual temperatures, and
- tracking step changes in set-point temperature with acceptable rise time, steady-state error, and overshoot.

Table 1 gives a more precise statement of specifications.
### Table 1. Temperature controller specifications

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set-point temperature entry</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>60–99°C</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>1°C</td>
<td></td>
</tr>
<tr>
<td><strong>Set-point temperature display</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>60–99°C</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>1°C</td>
<td></td>
</tr>
<tr>
<td><strong>Chamber temperature display</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>60–99°C</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>1°C</td>
<td></td>
</tr>
<tr>
<td><strong>Chamber temperature step response</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range (steady state)</td>
<td>60–99°C</td>
<td></td>
</tr>
<tr>
<td>Accuracy (steady state)</td>
<td>±1°C</td>
<td></td>
</tr>
<tr>
<td>Maximum overshoot</td>
<td>±1°C</td>
<td></td>
</tr>
<tr>
<td>Settling time (to ±1°C)</td>
<td>120 s</td>
<td></td>
</tr>
</tbody>
</table>

Although not explicitly a part of the specifications in Table 1, it was clear that the customer desired digital displays of set-point and actual temperatures, and that set-point temperature entry should be digital as well (as opposed to, say, through a potentiometer setting).

### 3 System Design

The requirements for digital temperature displays and set-point entry alone are enough to dictate that a microcontroller-based design is likely the most appropriate. Figure 2 shows a block diagram of the students’ design.

![Figure 2. Temperature controller hardware block diagram](image)

The microcontroller, a Motorola MC68HC705B16 (6805 for short), is the heart of the system. It accepts inputs from a simple four-key keypad which allow specification of the set-point temperature, and it displays both set-point and measured chamber temperatures using two-digit seven-segment LED displays controlled by a display driver. All these inputs and outputs are accommodated by parallel ports on the 6805. Chamber temperature is sensed using a pre-calibrated thermistor and input via one of the 6805’s analog-to-digital inputs. Finally, a pulse-width modulation (PWM) output on the 6805 is used to drive a relay which switches line power to the resistive heater off and on.

Figure 3 shows a more detailed schematic of the electronics and their interfacing to the 6805. The keypad, a Storm 3K041103, has four keys which are interfaced to pins PA0–PA3 of Port A, configured as inputs. One key functions as a mode switch. Two modes are supported: set mode and run mode. In set mode two of the other keys are used to specify the set-point temperature: one increments it and one decrements. The fourth key is unused at present. The LED displays are driven by a Harris Semiconductor ICM7212 display driver interfaced to pins PB0–PB6 of Port B, configured as outputs. The temperature-sensing thermistor drives, through a voltage divider, pin AN0 (one of eight analog inputs). Finally, pin PLMA (one of two PWM outputs) drives the heater relay.

![Figure 3. Schematic of microcontroller board](image)

Software on the 6805 implements the temperature control algorithm, maintains the temperature displays, and alters the set-point in response to keypad inputs. Because it is not complete at this writing, software will not be discussed in detail in this paper. The control algorithm in particular has not been determined, but it is likely to be a simple proportional
controller and certainly not more complex than a PID. Some control design issues will be discussed in Section 4, however.

4 The Design Process

Although essentially the project is just to build a thermostat, it presents many nice pedagogical opportunities. The knowledge and experience base of a senior engineering undergraduate are just enough to bring him or her to the brink of a solution to various aspects of the problem. Yet, in each case, real-world considerations complicate the situation significantly. Fortunately these complications are not insurmountable, and the result is a very beneficial design experience.

The remainder of this section looks at a few aspects of the problem which present the type of learning opportunity just described. Section 4.1 discusses some of the features of a simplified mathematical model of the thermal properties of the system and how it can be easily validated experimentally. Section 4.2 describes how realistic control algorithm designs can be arrived at using introductory concepts in control design. Section 4.3 points out some important deficiencies of such a simplified modeling/control design process and how they can be overcome through simulation. Finally, Section 4.4 gives an overview of some of the microcontroller-related design issues which arise and learning opportunities offered.

4.1 Mathematical Model

Lumped-element thermal systems are described in almost any introductory linear control systems text, and just this sort of model is applicable to the slide dryer problem.

Figure 4 shows a second-order lumped-element thermal model of the slide dryer. The state variables are the temperatures \( T_a \) of the air in the box and \( T_b \) of the box itself. The inputs to the system are the power output \( q(t) \) of the heater and the ambient temperature \( T_\infty \). \( m_a \) and \( m_b \) are the masses of the air and the box, respectively, and \( C_a \) and \( C_b \) their specific heats. \( \mu_1 \) and \( \mu_2 \) are heat transfer coefficients from the air to the box and from the box to the external world, respectively.

\[ \begin{align*}
T_a, m_a, C_a \\
T_b, m_b, C_b \\
\mu_1(T_a - T_b) \\
\mu_2(T_b - T_\infty)
\end{align*} \]

**Figure 4. Lumped-element thermal model**

It’s not hard to show that the (linearized) state equations corresponding to Figure 4 are

\[
\begin{align*}
m_a C_a \dot{T}_a &= \mu_1(T_a - T_b) \\
m_b C_b \dot{T}_b &= \mu_1(T_a - T_b) - (T_b - T_\infty)
\end{align*}
\]

Taking Laplace transforms of (1) and (2) and solving for \( T_a(s) \), which is the output of interest, gives the following open-loop model of the thermal system:

\[
T_a(s) = \frac{K(\tau_2s + 1)}{\Delta(s)}Q(s) + \frac{1}{\Delta(s)}T_\infty,
\]

where \( K \) is a constant and \( \Delta(s) \) is a second-order polynomial. \( K, \tau_2 \), and the coefficients of \( \Delta(s) \) are functions of the various parameters appearing in (1) and (2).

Of course the various parameters in (1) and (2) are completely unknown, but it’s not hard to show that, regardless of their values, \( \Delta(s) \) has two real zeros. Therefore the main transfer function of interest (which is the one from \( Q(s) \), since we’ll assume constant ambient temperature) can be written

\[
G_{aq}(s) = \frac{T_a(s)}{Q(s)} = \frac{K(\tau_2s + 1)}{(\tau_{p1}s + 1)(\tau_{p2}s + 1)}
\]

(3)

Moreover, it’s not too hard to show that \( 1/\tau_{p1} < 1/\tau_2 < 1/\tau_{p2} \), i.e., that the zero lies between the two poles. Both of these are excellent exercises for the student, and the result is the open-loop pole-zero diagram of Figure 5.

\[
-1/\tau_{p2} \quad -1/\tau_2 \quad -1/\tau_{p1}
\]

**Figure 5. Pole-zero diagram of \( G_{aq}(s) \)**

Obtaining a complete thermal model, then, is reduced to identifying the constant \( K \) and the three unknown time constants in (3). Four unknown parameters is quite a few, but simple experiments show that \( 1/\tau_{p1} < 1/\tau_2 < 1/\tau_{p2} \) so that \( \tau_2, \tau_{p2} \approx 0 \) are good approximations. Thus the open-loop system is essentially first-order and can therefore be written

\[
G_{aq}(s) = \frac{K}{\tau s + 1}
\]

(4)

(where the subscript \( p1 \) has been dropped).

Simple open-loop step response experiments show that, for a wide range of initial temperatures and heat inputs, \( K \approx 0.14 \text{ W}^{-1} \text{C}^{-1} \) and \( \tau \approx 295 \text{ s} \).

4.2 Control System Design

Using the first-order model of (4) for the open-loop transfer function \( G_{aq}(s) \) and assuming for the moment that linear control of the heater power output \( q(t) \) is possible, the block diagram of Figure 6 represents the closed-loop system. \( T_d(s) \) is

\textsuperscript{1}Of course the system is not actually linear, so the apparent parameter values vary with initial conditions and input magnitude. The effect on closed-loop performance is not too serious, but it gives the student a good idea of what nonlinearity means and how feedback tends to mitigate its effects.
the desired, or set-point, temperature, \( C(s) \) is the compensator transfer function, and \( Q(s) \) is the heater output in watts.

\[
T_d(s) 
\begin{array}{c}
\downarrow \quad C(s) \quad Q(s) \\
\uparrow
\end{array}
\frac{K}{\zeta s + 1} 
\rightarrow \quad T_a(s)
\]

**Figure 6. Simplified block diagram of the closed-loop system**

Given this simple situation, introductory linear control design tools such as the root locus method can be used to arrive at a \( C(s) \) which meets the step response requirements on rise time, steady-state error, and overshoot specified in Table 1. The upshot, of course, is that a proportional controller with sufficient gain can meet all specifications. Overshoot is impossible, and increasing gains decreases both steady-state error and rise time.

Unfortunately, sufficient gain to meet the specifications may require larger heat outputs than the heater is capable of producing. This was indeed the case for this system, and the result is that the rise time specification cannot be met. It is quite revealing to the student how useful such an oversimplified model, carefully arrived at, can be in determining overall performance limitations.

### 4.3 Simulation Model

Gross performance and its limitations can be determined using the simplified model of Figure 6, but there are a number of other aspects of the closed-loop system whose effects on performance are not so simply modeled. Chief among these are

- quantization error in analog-to-digital conversion of the measured temperature and
- the use of PWM to control the heater.

Both of these are nonlinear and time-varying effects, and the only practical way to study them is through simulation (or experiment, of course).

Figure 7 shows a Simulink™ block diagram of the closed-loop system which incorporates these effects. A/D converter quantization and saturation are modeled using standard Simulink quantizer and saturation blocks. Modeling PWM is more complicated and requires a custom S-function to represent it.

This simulation model has proven particularly useful in gauging the effects of varying the basic PWM parameters and hence selecting them appropriately. (i.e., the longer the period, the larger the temperature error PWM introduces. On the other hand, a long period is desirable to avoid excessive relay “chatter,” among other things.) PWM is often difficult for students to grasp, and the simulation model allows an exploration of its operation and effects which is quite revealing.

### 4.4 The Microcontroller

Simple closed-loop control, keypad reading, and display control are some of the classic applications of microcontrollers, and this project incorporates all three. It is therefore an excellent all-around exercise in microcontroller applications.

In addition, because the project is to produce an actual packaged prototype, it won’t do to use a simple evaluation board with the I/O pins jumpered to the target system. Instead, it’s necessary to develop a complete embedded application. This entails the choice of an appropriate part from the broad range offered in a typical microcontroller family and learning to use a fairly sophisticated development environment. Finally, a custom printed-circuit board for the microcontroller and peripherals must be designed and fabricated.

**Microcontroller Selection.** In view of existing local expertise, the Motorola line of microcontrollers was chosen for this project. Still, this does not narrow the choice down much. A fairly disciplined study of system requirements is necessary to specify which microcontroller, out of scores of variants, is required for the job. This is difficult for students, as they generally lack the experience and intuition needed as well as the perseverance to wade through manufacturers’ selection guides.

Part of the problem is in choosing methods for interfacing the various peripherals (e.g., what kind of display driver should be used?). A study of relevant Motorola application notes [2, 3, 4] proved very helpful in understanding what basic approaches are available, and what microcontroller/peripheral combinations should be considered.

The MC68HC705B16 was finally chosen on the basis of its available A/D inputs and PWM outputs as well as 24 digital I/O lines. In retrospect this is probably overkill, as only one A/D channel, one PWM channel, and 11 I/O pins are actually required (see Figure 3). The decision was made to err on the safe side because a complete development system specific to the chosen part was necessary, and the project budget did not permit a second such system to be purchased should the first prove inadequate.

**Microcontroller Application Development.** Breadboarding of the peripheral hardware, development of microcontroller software, and final debugging and testing of a custom printed-circuit board for the microcontroller and peripherals all require a development environment of some kind.
The choice of a development environment, like that of the microcontroller itself, can be bewildering and requires some faculty expertise. Motorola makes three grades of development environment ranging from simple evaluation boards (at around $100) to full-blown real-time in-circuit emulators (at more like $7500). The middle option was chosen for this project: the MMEVS, which consists of

- a **platform board** (which supports all 6805-family parts),
- an **emulator module** (specific to B-series parts), and
- a **cable and target head adapter** (package-specific).

Overall, the system costs about $900 and provides, with some limitations, in-circuit emulation capability. It also comes with the simple but sufficient software development environment RAPID [5].

Students find learning to use this type of system challenging, but the experience they gain in real-world microcontroller application development greatly exceeds the typical first-course experience using simple evaluation boards.

**Printed-Circuit Board.** The layout of a simple (though definitely not trivial) printed-circuit board is another practical learning opportunity presented by this project. The final board layout, with package outlines, is shown (at 50% of actual size) in Figure 8. The relative simplicity of the circuit makes manual placement and routing practical—in fact, it likely gives better results than automatic in an application like this—and the student is therefore exposed to fundamental issues of printed-circuit layout and basic design rules. The layout software used was the very nice package pcb, and the board was fabricated in-house with the aid of our staff electronics technician.

![Figure 8. Printed-circuit layout for microcontroller board](image)

**5 Conclusion**

The aim of this paper has been to describe an interdisciplinary, undergraduate engineering design project: a microcontroller-based temperature control system with digital set-point entry and set-point/actual temperature display. A particular design of such a system has been described, and a number of design issues which arise—from a variety of engineering disciplines—have been discussed. Resolution of these issues generally requires knowledge beyond that acquired in introductory courses, but realistically accessible to advance undergraduate students, especially with the advice and supervision of faculty.

Desirable features of the problem, from a pedagogical viewpoint, include the use of a microcontroller with simple peripherals, the opportunity to usefully apply introductory-level modeling of physical systems and design of closed-loop controls, and the need for relatively simple experimentation (for model validation) and simulation (for detailed performance prediction). Also desirable are some of the technology-related aspects of the problem including practical use of resistive heaters and temperature sensors (requiring knowledge of PWM and calibration techniques, respectively), microcontroller selection and use of development systems, and printed-circuit design.

**Acknowledgements**

The author would like to acknowledge the hard work, dedication, and ability shown by the students involved in this project: Mark Langsdorf, Matt Rall, Pam Rinehart, and David Schuchmann. It is their project, and credit for its success belongs to them.

**References**


